Contents

Part I
Plenary presentations

Space Decompositions and Solvers for Discontinuous Galerkin Methods

Blanca Ayuso de Dios[1] and Ludmil Zikatanov[2]

1 Introduction

The design and the analysis of efficient preconditioners for discontinuous Galerkin discretizations has been subject of intensive research in the last decade with efforts focused mainly on elliptic problems.

A standard point of view when studying most of the preconditioning and iterative solution strategies, in general, is associated with a particular *space decomposition*. From the classical theory of Lions [25, 30, 34], we know that, the choice of the space decomposition plays significant role in the construction and also in the convergence properties of the resulting preconditioners. For nonconfoming methods, domain decomposition and multigrid preconditioners have been analyzed by establishing connections with their respective conforming subspaces [10, 27]. In the case of DG methods, the discontinuous nature of the DG finite element spaces allows to introduce and study not only space splittings pertinent to the conforming methods but also consider new splittings which give rise to new techniques and ideas.

In most of the earlier works, relevant space splittings of the DG finite element space, were introduced via a domain decomposition. Overlapping additive Schwarz methods have been studied following the classical Schwarz theory for different DG schemes [21, 9, 20]. Contrary to the conforming case, additive (and multiplicative) Schwarz methods based on non-overlapping decomposition of the computational domain have been constructed and proven to be convergent for DG methods. For such type of preconditioners, novel features, which have no analog in the conforming case, arise. For both overlapping and non-overlapping Schwarz methods, the splittings are stable in the L^2-norm by construction and can be shown to be stable in the natural DG energy norm, with constants depending on the mesh sizes relative to the coarse and fine subspaces.

More sophisticated substructuring preconditioners have been studied recently for two dimensional elliptic Poisson problems. In [17, 18, 19, 1] non-overlapping BDDC, N-N, FETI-DP and BPS domain decomposition preconditioners are introduced and analyzed for a Nitsche-type approximation. BDDC preconditioners are studied in [15, 29] for IP-spectral and IP-hybridized methods. Also there, several different approaches have been considered and new theoretical tools have been introduced. And of course, the space splitting in which the preconditioner rely, comes

[1] Center for Uncertainty Quantification in Computational Science & Engineering, Division of Mathematics & Computer, Electrical and Mathematical Sciences & Engineering (CEMSE) King Abdullah University of Science and Technology, Kingdom of Saudi Arabia, e-mail: blanca.ayusodios@kaust.edu.sa [2] Department of Mathematics, The Pennsylvania State University, University Park, USA, e-mail: ludmil@psu.edu

always from domain decomposition. Starting directly with a splitting of the DG space, dictated by a hierarchy of meshes, multigrid methods have been proposed and analyzed in [22, 11]. A different approach was taken in [16] and [14, 13], to develop respectively, two-level and multilevel preconditioners for the Interior Penalty (IP) DG methods. A common idea behind these works is to use the fictitious/auxiliary spaces for which one knows how do develop a preconditioner. Such preconditioning techniques have already been applied in a wide range of problems in the conforming case.

The aforementioned auxiliary space preconditioners use error corrections from the conforming finite element space and they are certainly related to the a posteriori theory for DG methods [24]. In fact, the stable projections given in [24] provide the required tools for constructing and analyzing the convergence of these preconditioners including the case of non-conforming meshes.

A novel approach was taken in [8] where a natural decomposition of the linear DG finite element space was introduced. The components of the space decomposition are orthogonal in the inner product provided by the DG bilinear form. Such a splitting allows to devise efficient multilevel methods and uniform preconditioners and analyze these iterative schemes in a clean and transparent way. This seems to be the only approach available till now, to prove convergence for the solvers of the *non-symmetric* Interior Penalty methods. While the methodology has been applied to the lowest order DG space and conforming meshes, it is valid in two and three dimensions, and has already been adapted and extended to a larger family of problems: elliptic with jump coefficients [6]; linear elasticity [5]; and convection dominated problems corresponding to drift-diffusion models for transport of species [7].

We present here a brief overview of some of the domain and space decomposition techniques that comprise a set of key tools used in developing and analyzing solvers for DG methods. In Section 3 we focus on non-overlapping Schwarz domain decomposition methods. In Section 4 and 5 we present the two main classes of space decomposition methods commenting on their strengths and weaknesses.

2 Discontinuous Galerkin Methods

We consider the model problem for given data $f \in L^2(\Omega)$:

$$-\Delta u^* = f \quad \text{in } \Omega \qquad\qquad u^* = 0 \quad \text{on } \partial\Omega\,, \tag{1}$$

Here, $\Omega \subset \mathbb{R}^d$, $d = 2,3$ is a polygonal (polyhedral) domain. Let $\mathscr{T}_h$ be a shape-regular family of partitions of Ω into d-dimensional simplexes T (triangles if $d = 2$ and tetrahedrons if $d = 3$) and let $h = \max_{T\in\mathscr{T}_h} h_T$ with h_T denoting the diameter of T for each $T \in \mathscr{T}_h$. We denote by $\mathscr{E}_h^o$ and $\mathscr{E}_h^\partial$ the sets of all interior faces and boundary faces (edges in $d = 2$), respectively, and we set $\mathscr{E}_h = \mathscr{E}_h^o \cup \mathscr{E}_h^\partial$. Let V_h^{DG} denote the discontinuous finite element space defined by:

$$V_h^{DG} = \left\{ u \in L^2(\Omega) \,:\, u_{|_T} \,\in\, \mathbb{P}^\ell(T)\, \forall T \in \mathscr{T}_h \right\}, \tag{2}$$

where $\mathbb{P}^\ell(T)$ denotes the space of polynomials of degree at most ℓ on each T. We also define the conforming finite element space as $V_h^{\text{conf}} = V_h^{DG} \cap H_0^1(\Omega)$.
We define the *average* and *jump* trace operators. Let T^+ and T^- be two neighboring elements, and $\mathbf{n}^+$, $\mathbf{n}^-$ be their outward normal unit vectors, respectively ($\mathbf{n}^\pm = \mathbf{n}_{T^\pm}$). Let $\zeta^\pm$ and $\tau^\pm$ be the restriction of ζ and τ to $T^\pm$. We set:

$$\begin{array}{ll} 2\{\zeta\} = (\zeta^+ + \zeta^-), & [\![\zeta]\!] = \zeta^+\mathbf{n}^+ + \zeta^-\mathbf{n}^- \quad \text{on } E \in \mathscr{E}_h^o, \\ 2\{\tau\} \;= (\tau^+ + \tau^-), & [\![\tau]\!] = \tau^+ \cdot \mathbf{n}^+ + \tau^- \cdot \mathbf{n}^- \quad \text{on } E \in \mathscr{E}_h^o, \end{array}$$

$$[\![\zeta]\!] = \zeta\mathbf{n}, \qquad\qquad \{\tau\} = \tau \qquad \text{on } E \in \mathscr{E}_h^\partial. \tag{3}$$

We will also use the notation

$$(u,w)_{\mathscr{T}_h} = \sum_{T\in\mathscr{T}_h} \int_T uw dx \qquad \langle u,w\rangle_{\mathscr{E}_h} = \sum_{E\in\mathscr{E}_h} \int_E uw \quad \forall u,w, \in V_h^{DG}.$$

The approximation to the solution of (1) reads:

$$\text{Find} \quad u \in V_h^{DG} \quad \text{such that} \quad \mathscr{A}_h(u,w) = (f,w)_{\mathscr{T}_h}\,, \quad \forall w \in V_h^{DG}\,, \tag{4}$$

with $\mathscr{A}_h(\cdot,\cdot)$ the bilinear form corresponding to the Interior Penalty (IP) method (see [4]) defined by:

$$\mathscr{A}_h(u,w) = (\nabla u, \nabla w)_{\mathscr{T}_h} - \langle [\![u]\!], \{\nabla w\}\rangle_{\mathscr{E}_h} - \langle \{\nabla u\}, [\![w]\!]\rangle_{\mathscr{E}_h} + \langle S_h [\![u]\!], [\![w]\!]\rangle_{\mathscr{E}_h}\,, \tag{5}$$

where $S_h = \alpha_e \ell_e^2 h_e^{-1}$ with $\alpha_e \geq \alpha^* > 0$ for all $e \in \mathscr{E}_h$, h_e denotes the length of the edge e in $d = 2$ and the diameter of the face e in $d = 3$, and $\ell_e = \max\limits_{T^+\cap T^- = e} \{\ell_{T^+}, \ell_{T^-}\}$, with $\ell_{T^\pm}$ being the polynomial degree on $T^\pm$. Following [12], the above IP-biliear form can be re-written in terms of the weighed residual formulation:

$$\mathscr{A}_h(u,w) = (-\Delta u, w)_{\mathscr{T}_h} + \langle [\![\nabla u]\!], \{w\}\rangle_{\mathscr{E}_h^o} + \langle [\![u]\!], (S_h [\![w]\!] - \{\nabla w\})\rangle_{\mathscr{E}_h}\,. \tag{6}$$

Continuity and stability can be easily shown in the DG norm or in the induced $\|\cdot\|_{\mathscr{A}}$-norm, provided $\alpha_e \geq \alpha^* > 0$ is taken sufficiently large;

$$\begin{array}{lll} \text{Continuity:} & \mathscr{A}_h(u,w) \leq c_c \|u\|_{\mathscr{A}} \|w\|_{\mathscr{A}} & \forall u,w \in V_h^{DG} \\ \text{Coercivity:} & \mathscr{A}_h(u,u) \geq c_s \|u\|_{\mathscr{A}}^2 & \forall u \in V_h^{DG} \end{array} \tag{7}$$

3 Non-overlapping Domain Decomposition Schwarz methods

To define the non-overlapping preconditioners, we need to introduce some further notation. We denote by $\mathscr{T}_S$ the family of partitions of Ω into N non-overlapping subdomains $\Omega = \cup_{i=1}^N \Omega_i$. Together with $\mathscr{T}_S$, we let $\mathscr{T}_H$ and $\mathscr{T}_h$ be two families of

coarse and fine partitions, respectively, with mesh sizes H and h. The three families of partitions are assumed to be shape-regular and nested: $\mathscr{T}_S \subseteq \mathscr{T}_H \subseteq \mathscr{T}_h$.
Similarly as we did for $\mathscr{T}_h$ in Section 2, we define the skeleton and the corresponding sets of internal and boundary edges relative to the subdomain partition. In particular, for each subdomain $\Omega_i \in \mathscr{T}_S$ we define the sets of internal $\mathscr{E}_i^o = \{e \in \mathscr{E}_h : e \subset \Omega_i\}$ and boundary edges $\mathscr{E}_i^\partial = \{e \in \mathscr{E}_h : e \subset \partial\Omega_i\}$, and we set $\mathscr{E}_i = \mathscr{E}_i^o \cup \mathscr{E}_i^\partial$. Finally, we denote by Γ the collection of all interior edges that belong to the skeleton of the subdomain partition;

$$\Gamma = \bigcup_{i=1}^{N} \Gamma_i\,, \quad \text{with} \quad \Gamma_i = \{e \in \mathscr{E}_h^o : e \subset \partial\Omega_i\}.$$

The subdomain partition $\mathscr{T}_S$ induces a natural space splitting of the V^{DG} finite element space. More precisely, we have a local finite element subspace associated to each Ω_i for each $i = 1,\dots,S$, defined by

$$V_h^i = \{w \in V^{DG} : w \equiv 0 \quad \text{in } \subset \Omega \setminus \overline{\Omega_i}\}. \tag{8}$$

Let $\mathscr{I}_i^T : V_h^i \longrightarrow V_h^{DG}$ be the *prolongation* operator, defined as the standard inclusion operator that maps functions of V_h^i into V_h^{DG}. We denote by $\mathscr{I}_i$ the corresponding *restriction* operators defined (for each i) as the transpose of $\mathscr{I}_i^T$ with respect to the L^2–inner product. For vector-valued functions $\mathscr{I}_i^T$ and $\mathscr{I}_i$ are defined componentwise. Then the following splitting holds (orthogonal with respect to L^2-inner product):

$$V_h^{DG} = \mathscr{I}_1^T V_h^1 \oplus \mathscr{I}_2^T V_h^2 \oplus \ldots \oplus \mathscr{I}_N^T V_h^N\,. \tag{9}$$

LOCAL SOLVERS: Two types of local solvers have been considered:

(a). *Exact local solvers:* Following [21], the local solvers are defined as the restriction of the discrete bilinear form to the subspace V_i.

$$a_i(u_i, w_i) = \mathscr{A}_h(\mathscr{I}_i^T u_i, \mathscr{I}_i^T w_i) \qquad \forall\, u_i, w_i \in V_h^i \tag{10}$$

(b). *Inexact local solvers:* Following [2, 3] the local solvers are defined as the IP approximation to the original problem (1) but restricted to the subdomain Ω_i; i.e.,

$$-\Delta u_i^* = f|_{\Omega_i} \quad \text{in } \Omega_i, \qquad u_i^* = 0 \quad \text{on } \partial\Omega_i\,. \tag{11}$$

Then, the bilinear form can be written as:

$$\widehat{a}_i(u_i, w_i) = (-\Delta u_i, w_i)_{\mathscr{T}_h \cap \Omega_i} + \langle [\![\nabla u_i]\!], \{w_i\}\rangle_{\mathscr{E}_i^o} + \langle [\![u_i]\!], S_h [\![w_i]\!] - \{\nabla w_i\}\rangle_{\mathscr{E}_i}\,, \tag{12}$$

where in the above definition, edges on $\mathscr{E}_i^\partial$ are regarded as boundary edges (even those $e \in \mathscr{E}_i^\partial \setminus \partial\Omega_i$ so that $e \in \mathscr{E}_h^o$) and therefore the trace operators on such edges are defined as in (3).

Observe that, in a conforming framework, the definitions given in **(a)** and **(b)** would have given rise to exactly the same local solvers. The difference in the DG context,

originates from the distinct definition of the trace operators on boundary and internal edges and the fact that $e \in \mathscr{E}_i^{\partial} \setminus \partial\Omega_i$ is an interior edge for the global IP method (and so for (10)), but a boundary edge for (12). See [2, 3] for further details.

Let now $\mathbb{A}$ be the matrix representation of the operator associated to the global IP method (5), in some chosen basis (say nodal lagrange basis functions to fix ideas). We denote by $\mathbb{A}_i$ and $\widehat{\mathbb{A}}_i$ the matrix representation (stiffness matrix) of the operators associated to (10) and (12), respectively. At the algebraic level, a one-level Additive Schwarz preconditioner is then defined by $B_{add}^{one} = \sum_{i=1}^{S} \mathbb{I}_i^T \mathbb{S}_i^{-1} \mathbb{I}_i$ where $\mathbb{I}_i$ is the matrix representation of the restriction operator and $\mathbb{S}_i$ denotes here the matrix representation of the local solver; and can be chosen to be either $\mathbb{A}_i$ or $\widehat{\mathbb{A}}_i$. Notice however, that only for the choice $\mathbb{S}_i = \mathbb{A}_i$, the resulting one level additive Schwarz method B_{add}^{one} corresponds to the standard block jabobi preconditioner for the global stiffness matrix $\mathbb{A}$. This can be easily checked by noting that the definition (10) gives at the algebraic level $\mathbb{A}_i = \mathbb{I}_i \mathbb{A} \mathbb{I}_i^T$; that is, the matrices $\mathbb{A}_i$ are the principal submatrices of $\mathbb{A}$. In contrast, the one level additive Schwarz based on the choice $\mathbb{S}_i = \widehat{\mathbb{A}}_i$ cannot be obtained by starting directly from the algebraic structure of the global matrix $\mathbb{A}$; it would require further modifications of the prolongation and restriction operators.

On the other hand, in view of the possibility of considering (at least) these two definitions for the local solvers, a natural question arises. Namely, if the inexact local solvers (12) are approximating the original PDE restricted to the subdomain, *which continuous problem is approximated by the exact local solvers (10), if any.* By rewriting the bilinear form in the weighted residual formulation one easily obtains:

$$\begin{aligned} a_i(u_i, w_i) = (-\Delta u_i, w_i)_{\mathscr{T}_h \cap \Omega_i} &+ \langle [\![\nabla u_i]\!], \{w_i\} \rangle_{\mathscr{E}_i^o} \\ &+ \langle [\![u_i]\!], (S_h [\![w_i]\!] - \{\nabla w_i\}) \rangle_{\mathscr{E}_i^o \cup (\mathscr{E}_i^{\partial} \cap \partial\Omega)} \\ &+ \langle \tfrac{1}{2} \nabla u_i \cdot \mathbf{n} + S_h u_i, w_i \rangle_{\Gamma_i} - \langle u_i, \tfrac{1}{2} \nabla w_i \cdot \mathbf{n} \rangle_{\Gamma_i} \end{aligned} \tag{13}$$

The terms on the first and second lines are easy to recognize, the first imposes the PDE on each element; the second is the consistency term and the terms in the second line ensure stability and symmetry. As regards those in the last line, the first term is imposing the boundary condition on Γ_i (the part of $\partial\Omega_i \setminus \partial\Omega$). The second term, could be regarded as an artifact to ensure the symmetry of the method. Then, one can write the continuous problem

$$\begin{cases} -\Delta u_i^* &= f|_{\Omega_i} & \text{in } \Omega_i, \\ u_i^* &= 0 & \text{on } \partial\Omega_i \cap \partial\Omega\,, \\ \frac{1}{2} \frac{\partial u_i^*}{\partial n_i} + S_h u_i^* &= 0 & \text{on } \Gamma_i\,. \end{cases} \tag{14}$$

This implies that the exact local solvers for the IP method (and in general for most DG methods) are approximating the original problem but with transmission Robin conditions. And as $h \to 0$ the method enforces $u_i^* = 0$ on Γ_i. Whether such interface boundary conditions are optimal or could be further tuned to improve the convergence properties of the classical Schwarz methods is a subject of current research.

Optimization of the Schwarz methods with respect to the interface boundary conditions has been recently studied in [23]. The final ingredient needed to define the two-level Schwarz method is the coarse solver.
COARSE SOLVER: Let $V_c := V_H^{DG}$ be the coarse space and let $a_c : V_c \times V_c \longrightarrow \mathbb{R}$ be the coarse solver defined by [21, 2, 3]:

$$a_c(u_c, w_c) = \mathscr{A}_h(\mathscr{I}_c^T u_c, \mathscr{I}_c^T w_c) \qquad \forall u_c, w_c \in V_c \tag{15}$$

where $\mathscr{I}_c^T : V_c \longrightarrow V_h^{DG}$ is the prolongation operator, defined as the standard inclusion. Notice that with this definition, the corresponding matrices do indeed satisfy the Galerkin property: $\mathbb{A} = \mathbb{I}_c^T \mathbb{A}_c \mathbb{I}_c$, but should be noted that unlike in a conforming framework $a_c(u_c, w_c) \neq \mathscr{A}_H(u_c, w_c)$. A two level Schwarz preconditioner can then be defined:

$$\mathbb{B}_{add} = \sum_{i=1}^{S} \mathbb{I}_i^T \mathbb{S}_i^{-1} \mathbb{I}_i + \mathbb{I}_c^T \mathbb{A}_c^{-1} \mathbb{I}_c \tag{16}$$

It is also possible to define the coarse solver as IP approximation (with the partition $\mathscr{T}_H$ and the coarse space V_c) to the orginal problem (i.e., as $\mathscr{A}_H(u_c, w_c)$). However with such definition, the Galerkin property is lost and in order to ensure scalability of the resulting two level Schwarz preconditioner, more sophisticated prologation and restriction operators are required [9].

Let now B^{-1} denote the inverse operator associated to the two level preconditioner (16). To analyze the convergence properties of the resulting preconditioner one needs to characterize the dependence of the constants C_1 and C_0 in

$$C_1 \mathscr{A}_h(w, w) \leq (B^{-1} w, w) \leq C_0^2 \mathscr{A}_h(w, w) \qquad \forall w \in V_h^{DG} \tag{17}$$

The condition number of the preconditioned matrix $\mathbb{B}\mathbb{A}$ is then C_0^2/C_1. The proof of (17) is often guided by Lions lemma (for a proof see [32], [31], [34, Lemma 2.4]), which tells that the preconditioner can be written as

$$(B^{-1} w, w) := \inf_{\substack{w_i \in V^i \\ w_c + \sum_i w_i = w}} \left(a_c(w_c, w_c) + \sum_i \mathscr{R}_i(w_i, w_i) \right), \tag{18}$$

where we have denoted by $\mathscr{R}_i(\cdot, \cdot)$ the *approximate (or exact) subspace solver* on V^i.

4 Ficticious Space and Auxiliary Space Methods

Ficticious Space Lemma was originally introduced by Nepomnyaschikh in [26], and further used for developing and analyzing multilevel preconditioners for nonconforming approximations in [27] and for conforming methods with nonconforming

meshes in [33]. There are two main ingredients to construct a fictitious space preconditioner for the operator $A : V_h^{DG} \longrightarrow V_h^{DG}$ associated to the bilinear form (5).

(1) A fictitious space $\overline{V}$, and an symmetric positive definite operator $\overline{A} : \overline{V} \longrightarrow \overline{V}$ associated with some $\overline{\mathscr{A}}(\cdot,\cdot) : \overline{V} \times \overline{V} \longrightarrow \mathbb{R}$.
(2) A continuous, linear and surjective mapping $\Pi : \overline{V} \to V_h^{DG}$

The fictitious space preconditioner B is then defined as

$$B = \Pi \circ \overline{A}^{-1} \circ \Pi^* : V_h^{DG} \to V_h^{DG}. \tag{19}$$

The convergence properties of the preconditioner B depend on the choice of the fictitious space $\overline{V}$ and ficticious operator $\overline{A}$. Typically, one chooses a fictitious pair $(\overline{V}, \overline{A})$ for which it is simpler to construct a preconditioner. The analysis of such methods is done via the *Fictitious space lemma* [26], which states that if Π has a bounded (in energy norm) right inverse and is stable in $\overline{A}$ norm, then B is equivalent to A (in the sense that they satisfy a corresponding (17)) with constants of equivalence (C_1 and C_0^2) depending on the stability and invertibility of Π. The auxiliary space idea, comes from the observation (see [33]) that a *surjective* Π is easy to construct for the choice $\overline{V} = V_h^{DG} \times W$ for some space W (the factor V_h^{DG} in the product plays a crucial role).

One natural approach in constructing such preconditioners for DG discretizations is via subspace splitting which uses the corresponding conforming space as the component W; that is $\overline{V} = V_h^{DG} \times V_{\tilde{h}}^{\text{conf}}$, with $W := V_{\tilde{h}}^{\text{conf}}$ denoting the conforming finite element space with $\tilde{h}$ chosen $\tilde{h} \geq h$. This is natural because one expects that the smooth error (with small energy) is in this space. Then, for the auxiliary preconditioner $\overline{A}^{-1}$ one can choose his/her favourite solver in $V_{\tilde{h}}^{\text{conf}}$. Preconditioners based on such splittings are found in [16] and [14], and more recently in [13, 15]. Two-level methods based on three different splittings of the DG space are given in [16]. In [14], an auxiliary space preconditioner is proposed (and analyzed) for IP discretizations with non-conforming meshes and hanging nodes. This auxiliary space approach has been recently extended and used for designing multilevel preconditioners in [13] for the IP method with arbitrary polynomial degree. The results from [13] are further used for constructing a BDDC preconditioner for such discretizations in [15].

We wish to point out that for the IP method such decompositions were already known in the area of adaptivity and a posteriori error analysis for DG methods. The following important decomposition is implicitly contained in In the seminal work [24]:

$$V_h^{DG} = V_h^{\text{conf}} \oplus E_h, \tag{20}$$

where $E_h = (V_h^{\text{conf}})^{\perp}$ refers to the complementary space of V_h^{conf} in V^{DG} (orthogonal with respect to the corresponding energy inner product). In fact, an explicit construction of an interpolation operator $I_h : V_h^{DG} \longrightarrow V_h^{\text{conf}}$ is provided, on simplicial meshes, even in case of hanging nodes, which is stable in the energy norm, and therefore can be used as a component in constructing a stable surjective Π in the design of an auxiliary space preconditioner.

The analysis of the auxiliary space preconditioners using the conforming method as a component of the space decomposition is carried out in a standard fashion by introducing stable and accurate interpolation operators (see e.g. [14] or [16] for such constructions). Alternatively, at least for the *h*-version, one may adapt and use the framework developed in [24] to analyse the properties of these preconditioners.

5 Orthogonal space splittings in a nutshell

The approach we present now has been developed in [8] for developing uniform solvers for the family of IP discretizations, including non-symmetric schemes. It could be seen as a clever change of basis which allows for special decompositions of the DG space. The ideas work in dimensions $d = 2,3$ and are based on a natural splitting of the linear DG FE space on simplicial meshes with no-hanging nodes. Therefore, in all what follows V^{DG} stands for the linear approximation space; i.e., $\ell = 1$. Furthermore, to ease the presentation, we drop the subscript h from the finite element space and the bilinear form, so $\mathscr{A}(\cdot,\cdot) = \mathscr{A}_h(\cdot,\cdot)$. For multilevel considerations see for instance [6]. To introduce the space splitting we first introduce some notation.

Together with the IP bilinear form $\mathscr{A}(\cdot,\cdot)$, we also consider the bilinear form that results by computing all the integrals in (5) with the mid-point quadrature rule, known as weakly penalized or IP-0 method:

$$\mathscr{A}_0(u,w) = (-\Delta u, w)_{\mathscr{T}_h} + \langle [\![\nabla u]\!], \{w\}\rangle_{\mathscr{E}_h^o} + \langle \mathscr{P}_E^0([\![u]\!]), S_h[\![w]\!] - \{\nabla w\}\rangle_{\mathscr{E}_h}\,, \quad (21)$$

where, for each $e \in \mathscr{E}_h$, let $\mathscr{P}_e^0 : L^2(e) \longrightarrow \mathbb{P}^0(e)$ is the L^2-orthogonal projection onto the constants on that edge defined by:

$$\mathscr{P}_e^0(u) := \frac{1}{|e|}\int_e u, \qquad \forall\, u \in L^2(e). \quad (22)$$

We define the following two subspaces of V^{DG}

$$V^{CR} := \{v \in V^{DG} \;:\; \mathscr{P}_e^0([\![v]\!]) = 0 \;\forall\, e \in \mathscr{E}_h^o\} \quad (23)$$

$$\mathscr{Z} := \{z \in V^{DG} \;:\; \mathscr{P}_e^0(\{z\}) = 0 \;\forall\, e \in \mathscr{E}_h\} \quad (24)$$

The first one is the well known lowest order Crouziex-Raviart finite element space. The above subspaces can be seen to be complementary to each other, and in fact it is easy to prove that

$$V^{DG} = V^{CR} \oplus \mathscr{Z}\,. \quad (25)$$

Notice that the explicit characterization of the subspaces allows to provide basis for both spaces. (See Fig. 1).

A key property satisfied by the space decomposition (25) is that the two subpaces are orthogonal in the enegy norm defined by $\mathscr{A}_0(\cdot,\cdot)$. In fact it can be easily shown

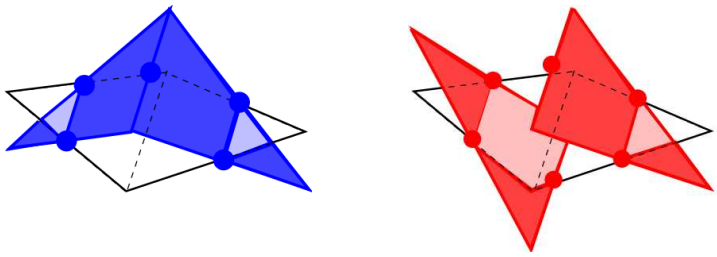


Fig. 1 Basis functions (associated to an edge) for the Crouziex Raviart space (left figure) and the $\mathscr{Z}$ space (right figure)

using (21) and the definition of the spaces (23) and (24) that

$$\mathscr{A}_0(v,z) = \mathscr{A}_0(z,v) = 0 \qquad \forall v \in V^{CR}\,,\, z \in \mathscr{Z}\,. \tag{26}$$

This already suggest that by perfoming a *change of basis* of the standard Lagrange basis for V^{DG} to the ones in V^{CR} and $\mathscr{Z}$, the stiffness matrix representation of $\mathbb{A}_0$ in the new basis have a block diagonal structure. Therefore, for the IP-0 method the following algorithm is an exact solver:

Algorithm 1: Let u_0 be a given initial guess. For $k \geq 0$, and given $u_k = z_k + v_k$, the next iterate $u_{k+1} = z_{k+1} + v_{k+1}$ is defined via the two steps:

1. Solve $\mathscr{A}_0(z_{k+1}, \psi^z) = (f, \psi^z)_{\mathscr{T}_h} \quad \forall\, \psi^z \in \mathscr{Z}$.
2. Solve $\mathscr{A}_0(v_{k+1}, \varphi) = (f, \varphi)_{\mathscr{T}_h} \quad \forall\, \varphi \in V^{CR}$.

Notice that algorithm 1 requires two solutions of smaller problems: one solution in $\mathscr{Z}$-space (step 1 of the algorithm 1), and one solution in V^{CR}-space (step 2 of algorithm 1). As we show next, the solution of the subproblems on $\mathscr{Z}$ and on V^{CR} can be done efficiently.

SOLUTION IN THE $\mathscr{Z}$-SPACE: The functions in $\mathscr{Z}$ have non-zero jump on every edge, which suggest the high oscillatory nature of its functions. Using the definition of the space, the following useful property (Poincare-type inequality) can be shown:

Lemma 1. *Let $\mathscr{Z}$ be the space defined in (24).*

$$h^{-2}\|z\|^2_{0,\mathscr{T}_h} \lesssim \mathscr{A}_0(z,z) \lesssim h^{-2}\|z\|^2_{0,\mathscr{T}_h}\,, \qquad \forall z \in \mathscr{Z}$$

By virtue of this lemma it follows that the condition number (denoted by κ) of the block matrix associated to the restriction of $\mathscr{A}_0(\cdot,\cdot)$ to the subspace $\mathscr{Z}$, say $\mathbb{A}_0^{zz}$, satisfies $\kappa(\mathbb{A}_0^{zz}) = O(1)$ and it is independent of the mesh size. Therefore, efficient solver for the problem in $\mathscr{Z}$ is the Conjugate Gradient (CG) method with a simple diagonal preconditioner.

SOLUTION IN V^{CR}: The restriction of $\mathscr{A}_0(\cdot,\cdot)$ to the V^{CR} subspace gives the well-known Crouziex-Raviart approximation method for (1) ;

$$\mathscr{A}_0(v,\varphi) = (\nabla v, \nabla \varphi)_{\mathscr{T}_h} = \sum_{T\in\mathscr{T}_h} (\nabla v, \nabla \varphi)_T \quad \forall\, v, \varphi \in V^{CR}\,, \tag{27}$$

Therefore, it is enough to resort to any of the solvers that have been already developed, for instance [10, 27, 28].

So far, an exact solver has been constructed in a simple and clean way for the IP-0 method. A last ingredient is needed to provide uniformly convergent solvers for the IP method (5) and it is formulated in next Lemma:

Lemma 2. *Let $\mathscr{A}(\cdot,\cdot)$ and $\mathscr{A}_0(\cdot,\cdot)$ be the bilinear forms of the IIPG method defined in (5) and (21). Then, there exist $c_2 > 0$ depending only on the shape regularity of $\mathscr{T}_h$ and $c_0 > 0$ depending also on the penalty parameter α such that*

$$c_2\mathscr{A}_0(u,u) \lesssim \mathscr{A}(u,u) \leq c_0\mathscr{A}_0(u,u) \quad \forall u \in V^{DG}. \tag{28}$$

The above result establishes the *spectral equivalence* between $\mathscr{A}_0(\cdot,\cdot)$ and $\mathscr{A}(\cdot,\cdot)$. Therefore, in terms of solution techniques, a uniform preconditioner for the IP-0 method, already provides a uniform preconditioner for the IP method.

These ideas and new framework, have been already extended and adapted for designing and analyzing solvers for other problems:

• In [6] the case of second order elliptic problems with large *jumps in the diffusion coefficient* is considered. In a first step, the space splitting (25) needs to be modified to account for the jumps in the coefficient, while still being orthogonal with respect to the corresponding $\mathscr{A}_0(\cdot,\cdot)$-induced norm. The choice of a robust method for approximating the continuous problem (definition of the relevant $\mathscr{A}(\cdot,\cdot)$ bilinear form) allows to guarantee that the corresponding spectral equivalence property (28) holds with constants c_0, c_2 independent of the mesh size and the *jumping coefficient*.

• In [5] efficient solvers are analyzed for IP approximations of *linear elasticity problems*, considering all cases: the pure displacement, the mixed and the traction free problems. The last two cases pose some extra pitfalls in the analysis since the spectral equivalence property (28) does not hold in those cases. In spite of that, the ideas can still be used to construct block preconditioners (guided by the algebraic structure of $\mathscr{A}_0(\cdot,\cdot)$ due to the orthogonality) and prove uniform convergence.

• In [7] it is shown how to construct an efficient solver for the solution of the linear system that arise from a DG discretization of a convection-diffusion problem, in the convection dominated regime. The problem is relevant in semiconductor applications. In this case, the original method is a non-symmetric exponentially fitted IP weakly-penalized.

Acknowledgements B. Ayuso de Dios thanks R. Hiptmair (ETH) for raising up the issue on the use of Auxiliary space techniques for preconditioning DG methods, at the DD21 meeting. First author has been partially supported by MINECO grant MTM2011-27739-C04-04 and GENCAT 2009SGR-345.

References

1. Antonietti, P., Ayuso de Dios, B., Bertoluzza, S., Penacchio, M.: BPS preconditioners for an hp Nitsche discretization. Tech. rep., IMATI-CNR, Pavia (17-PV-12/16/0) (2012). ArXiv:1301.3175
2. Antonietti, P.F., Ayuso, B.: Schwarz domain decomposition preconditioners for discontinuous Galerkin approximations of elliptic problems: non-overlapping case. Math. Model. Numer. Anal. **41**(1), 21–54 (2007)
3. Antonietti, P.F., Ayuso, B.: Multiplicative Schwarz methods for discontinuous Galerkin approximations of elliptic problems. Math. Model. Numer. Anal. **42**(3), 443–469 (2008)
4. Arnold, D.N.: An interior penalty finite element method with discontinuous elements. SIAM J. Numer. Anal. **19**(4), 742–760 (1982)
5. Ayuso de Dios, B., Georgiev, I., Kraus, J., Zikatanov, L.: A subspace correction methods for discontinuous Galerkin methods discretizations for linear elasticity equations. ESAIM: Math. Model. Numer. Anal. **47**(5), 1315–1333 (2013). URL `http://dx.doi.org/10.1051/m2an/2013070`
6. Ayuso de Dios, B., Holst, M., Zhu, Y., Zikatanov, L.: Multilevel preconditioners for discontinuous Galerkin approximations of elliptic problems with jump coefficients. Math. Comp. DOI:10.1090/S0025-5718-2013-02760-3 (Published online on October 30, 2013).
7. Ayuso de Dios, B., Lombardi, A., Pietra, P., Zikatanov, L.: A block solver for the exponentially fitted IIPG-0 method. In: Twenty International Conference on Domain Decomposition, *Lecture Notes in Computational Science and Engineering*, vol. 91, pp. 247–255. Springer–Verlag (2013). URL `http://dx.doi.org/10.1007/978-3-642-35275-1_28`
8. Ayuso de Dios, B., Zikatanov, L.: Uniformly convergent iterative methods for discontinuous Galerkin discretizations. J. Sci. Comput. **40**(1-3), 4–36 (2009)
9. Barker, A., Brenner, S., Park, E.H., Sung, L.: Two-level additive Schwarz preconditioners for a weakly over-penalized symmetric interior penalty method. J. Sci. Comput. (2010)
10. Brenner, S.C.: Two-level additive Schwarz preconditioners for nonconforming finite element methods. Math. Comp. **65**(215), 897–921 (1996)
11. Brenner, S.C., Zhao, J.: Convergence of multigrid algorithms for interior penalty methods. Appl. Numer. Anal. Comput. Math. **2**(1), 3–18 (2005)
12. Brezzi, F., Cockburn, B., Marini, L.D., Süli, E.: Stabilization mechanisms in discontinuous Galerkin finite element methods. Comput. Methods Appl. Mech. Engrg. **195**(25-28), 3293–3310 (2006)
13. Brix, K., Campos Pinto, M., Canuto, C., Dahmen, W.: Multilevel preconditioning of discontinuous Galerkin spectral element methods. Part I: Geometrically conforming meshes. Tech. rep., IGPM Preprint, RWTH Aachen (2013). ArXiv:1301.6768
14. Brix, K., Campos Pinto, M., Dahmen, W.: A multilevel preconditioner for the interior penalty discontinuous Galerkin method. SIAM J. Numer. Anal. **46**(5), 2742–2768 (2008)
15. Canuto, C., Pavarino, L., Pieri, A.: BDDC preconditioners for Continuous and Discontinuous Galerkin methods using spectral/hp elements with variable local polynomial degree. IMA J. Numer. Anal. (2013). DOI: 10.1093/imanum/drt037
16. Dobrev, V.A., Lazarov, R.D., Vassilevski, P.S., Zikatanov, L.T.: Two-level preconditioning of discontinuous Galerkin approximations of second-order elliptic equations. Numer. Linear Algebra Appl. **13(**9), 753–770 (2006)
17. Dryja, M., Galvis, J., Sarkis, M.: BDDC methods for discontinuous Galerkin discretization of elliptic problems. J. Complexity **23**(4-6), 715–739 (2007)
18. Dryja, M., Galvis, J., Sarkis, M.: Neumann-Neumann methods for a DG discretization of elliptic problems with discontinuous coefficients on geometrically nonconforming substructures. Numer. Methods Partial Differential Equations **28**(4), 1194–1226 (2012)
19. Dryja, M., Sarkis, M.: FETI-DP method for a Composite Finite Element and Discontinuous Galerkin Method. SIAM J. Numer. Anal. **51**(1), 400–422 (2013)
20. E. T. Chung, H.H.K., Widlund, O.B.: Two-level overlapping Schwarz algorithms for a staggered discontinuous Galerkin method. SIAM J. Numer. Anal. **51**(1), 47–67 (2013)

21. Feng, X., Karakashian, O.A.: Two-level additive Schwarz methods for a discontinuous Galerkin approximation of second order elliptic problems. SIAM J. Numer. Anal. **39**(4), 1343–1365 (electronic) (2001)
22. Gopalakrishnan, J., Kanschat, G.: A multilevel discontinuous Galerkin method. Numer. Math. **95**(3), 527–550 (2003)
23. Hajian, S., Gander, M.: Block Jacobi for discontinuous Galerkin discretizations: no ordinary Schwarz methods. In *Domain Decomposition Methods in Science and Engineering XXI*, Lect. Notes Comput. Sci. Eng. Springer, same volume, 2014.
24. Karakashian, O.A., Pascal, F.: A posteriori error estimates for a discontinuous Galerkin approximation of second-order elliptic problems. SIAM J. Numer. Anal. **41**(6), 2374–2399 (electronic) (2003)
25. Lions, P.L.: On the Schwarz alternating method. I. In: First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987), pp. 1–42. SIAM, Philadelphia, PA (1988)
26. Nepomnyaschikh, S.V.: Mesh theorems on traces, normalizations of function traces and their inversion. Soviet J. Numer. Anal. Math. Modelling **6**(3), 223–242 (1991)
27. Oswald, P.: Preconditioners for nonconforming discretizations. Math. Comp. **65**(215), 923–941 (1996)
28. Sarkis, M.: Nonstandard coarse spaces and Schwarz methods for elliptic problems with discontinuous coefficients using non-conforming elements. Numer. Math. **77**(3), 383–406 (1997)
29. Schöberl, J., Lehrenfeld, C.: Domain decomposition preconditioning for high order hybrid discontinuous Galerkin methods on tetrahedral meshes. In: Advanced finite element methods and applications, *Lect. Notes Appl. Comput. Mech.*, vol. 66, pp. 27–56. Springer (2013)
30. Toselli, A., Widlund, O.: Domain Decomposition Methods: Algorithms and Theory. Springer Series in Computational Mathematics (2005)
31. Widlund, O.B.: Some Schwarz methods for symmetric and nonsymmetric elliptic problems. In: Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations. SIAM (1992)
32. Xu, J.: Iterative methods by space decomposition and subspace correction. SIAM Rev. **34**(4), 581–613 (1992)
33. Xu, J.: The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. Computing **56**(3), 215–235 (1996). International GAMM-Workshop on Multi-level Methods (Meisdorf, 1994)
34. Xu, J., Zikatanov, L.: The method of alternating projections and the method of subspace corrections in Hilbert space. J. Amer. Math. Soc. **15**(3), 573–597 (2002)

A finite element method for particulate flow

Eberhard Bänsch[1] and Rodolphe Prignitz[1]

1 Introduction

Particulate flow, i.e. the flow of a carrier fluid loaded with particles, plays an important role in many technical applications. Let us just mention reactors, fluidized beds, production of nano particles and many more. There exists a hierarchy of models how to describe the particulate phase and how to describe the interaction between particles and fluid. For a comprehensive list of references we refer to the articles of Esmaelli & Tryggvason [6] and Hu [12].

For certain applications it is mandatory to describe the fluid–particle interaction and also a possible particle-particle interaction in full detail without simplified parametrizations. Computational methods based on such full models are called *direct numerical simulations*.

One of the most important points in simulating particulate flow is the numerical representation of the particles' geometry. In Feng et al. and Johnson & Tezduyar [7, 13] a remeshing technique was used to explicitly follow the geometry in time; Wan and Turek [22] introduced a mesh deformation technique and Glowinski et al. [9] used Lagrange multipliers on regular grids. Also immersed boundary methods are very popular, for example LeVeque & Li and Veeramani et al. [14, 20]. Distributed Lagrange multipliers to account for the stress boundary condition are used in Bönisch & Heuveline and Bönisch et al. [5, 4]. In Maury [16] a projection based method was already introduced, still following explicitly the geometry, thus requiring remeshing.

Analytical results regarding existence, uniqueness and qualitative behavior of solutions can be found for instance in Galdi and Serre [8, 19].

The approach presented here is based on the *one domain approach* by [19, 9], but differs from the above mentioned articles in one or several aspects, since it

- does not require an explicit meshing of the particles' domain;
- does not need an explicit evaluation of forces;
- uses a *subspace projection method* to account for the constraint of rigid body motion within the particles, thus avoiding a saddle point problem for this constraint;
- uses time dependent adaptively refined meshes to provide the necessary geometric resolution.

[1] U of Erlangen, Cauerstr. 11, 91058 Erlangen, Germany, e-mail: {baensch}{prignitz}@math.fau.de

It turns out that this novel method is therefore easy to implement (only few modules have to be added to an existing standard software) and rather efficient. A more detailed presentation can be found in [17].

2 Mathematical Formulation

2.1 Model

In this section we introduce the mathematical model for particulate flows. For ease of presentation we restrict ourselves to the case of a single particle. The extension to more particles is straightforward, simply by adding an index. The model also holds for the $2d$-case, one just has to adapt the definition of the cross-product involved in the equations.

Denote by $\Omega(t) \subset \mathbb{R}^3$ the time-dependent domain occupied by an incompressible, Newtonian fluid with velocity u and pressure p. Its motion is described by the incompressible Navier-Stokes equations. A homogeneous no-slip condition is prescribed on the outer boundary Γ_D.

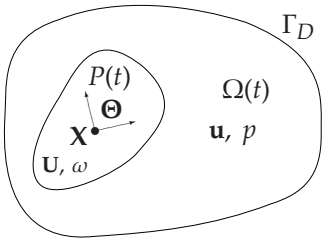


Fig. 1 Particle $P(t)$ of arbitrary shape inside the fluid domain $\Omega(t)$.

$P(t) \subset \mathbb{R}^3$ is the time-dependent domain of a rigid particle, with its center of mass given by $X = \frac{1}{|P(t)|}\int_{P(t)} x\,dx$, while $r = x - X$ is its relative coordinate. The particle's motion, being a rigid body motion, is governed by Newton's law, describing values for the translational and angular velocities U, ω, respectively, and the position X. The orientation in space is given by a complete system of orthogonal unit vectors who's coordinates are denoted by Θ. Since the particle is impermeable, we assume $\Omega(t) \cap P(t) = \emptyset$ for all times $t > 0$. Finally we assume (for simplicity) that the whole volume $\Omega_c = \Omega(t) \cup P(t) \cup \partial P(t)$ is time independent. See also Fig. 2.1 for a sketch of the situation.

The motions of fluid and particle are coupled on one hand by the *no-slip-condition* on the particle boundary Eq. (4) below and and on the other hand by the stress and pressure forces of the fluid acting on the particle (in the right hand sides of Eq. (5)). The mathematical model consists of a coupled system of partial

differential equations (PDE) for u, p and of ordinary differential equations (ODE) for U, ω, X and Θ reading in non-dimensional form

$$\partial_t u + (u \cdot \nabla) u - \nabla \cdot \left(\frac{1}{\mathrm{Re}} \mathrm{D}[u] - p\mathbb{I} \right) = 0 \quad \text{in } \Omega(t), \tag{1}$$

$$\nabla \cdot u = 0 \quad \text{in } \Omega(t), \tag{2}$$

$$u = 0 \quad \text{on } \Gamma_D, \tag{3}$$

$$u = U + \omega \times r \quad \text{on } \partial P(t), \tag{4}$$

$$M\dot{U} = F - \int_{\partial P(t)} \sigma n \, ds, \quad I\dot{\omega} + \omega \times (I\omega) = - \int_{\partial P(t)} r \times \sigma n \, ds, \tag{5}$$

$$\dot{X} = U, \quad \dot{\Theta} = \mathrm{R}[\omega]\Theta. \tag{6}$$

The system has to be closed by appropriate initial conditions. Here, Re is the Reynolds number, M and I the mass and inertia tensors, respectively; $\sigma := \frac{1}{\mathrm{Re}} \mathrm{D}[u] - p\mathbb{I}$ is the stress tensor, where $\mathrm{D}[\cdot]$ is the deformation tensor $\mathrm{D}[u]_{i,j} = \partial_j u_i + \partial_i u_j$. F describes an external force acting on the particle like gravity, particle-particle (in case of more than 1 particle) or particle-wall interaction. $\mathrm{R}[\cdot]$ is the cross-product operator.

2.2 Weak formulation

Following the idea and presentation in [9] a weak formulation of the system Eqs. (1)-(6) is presented. This formulation is instrumental for deriving our numerical method in the next section. Define

$$H_c(\Omega_c) = \left\{ (v, V, \xi) \,\middle|\, v \in \left(H^1(\Omega_c)\right)^3, V \in \mathbb{R}^3, \xi \in \mathbb{R}^3, v = 0 \text{ on } \Gamma_D, v = V + \xi \times r \text{ in } P(t) \right\}. \tag{7}$$

Note that by the above definition the velocity v in $H_c(\Omega_c)$ is defined on the combined domain Ω_c and is restricted to the rigid body velocity $V + \xi \times r$ inside the particle. For a shorter notation we introduce the bi- and trilinear forms

$$m(u, v) = \int_{\Omega_c} u \cdot v \, dx, \tag{8}$$

$$s(u, v) = \frac{1}{2\mathrm{Re}} \int_{\Omega_c} \mathrm{D}[u] : \mathrm{D}[v] \, dx, \tag{9}$$

$$k(w; u, v) = \int_{\Omega_c} (w \cdot \nabla) u \cdot v \, dx, \tag{10}$$

$$b(q, v) = \int_{\Omega_c} q \nabla \cdot v \, dx, \tag{11}$$

and the variable $\beta = 1 - \alpha$. Then Eqs. (1)–(6) can be compactly written as:

Find $(\mathfrak{u},p)$ with $\mathfrak{u}(t)\in H_c(\Omega_c)$, $p(t)\in L^2_0(\Omega_c)$ such that for all $(\mathfrak{v},q)\in \big(H_c(\Omega_c)\times L^2_0(\Omega_c)\big)$

$$\begin{aligned} m(\dot u,v)+k(u;v,u)+s(u,v)-b(p,v)+&\\ \beta M\dot U\cdot V+\beta\left(I\dot\omega+\omega\times(I\omega)\right)\cdot\xi &= F\cdot V, && (12)\\ b(q,u) &= 0, && (13)\\ \dot X &= U, && (14)\\ \dot\Theta &= \mathrm{R}[\omega]\,\Theta. && (15) \end{aligned}$$

Eqs. (12) and (13) are called the *combined* Navier-Stokes equations. The time dependence of $\Omega(t)$ and $P(t)$ is now completely coded in the time dependent definition of $H_c(\Omega_c)$.

3 Numerical Method

The numerical scheme to solve the weak problem Eqs. (12)–(15) derived in the previous section consists of the following six points:

1. splitting scheme to decouple the unknowns;
2. a pressure correction projection scheme based on a BDF2 method to efficiently solve the combined Navier-Stokes equations;
3. subspace projection to incorporate the restrictions given by the function space $H_c(\Omega_c)$;
4. adaptivity in space;
5. preconditioning;
6. Barnes-Hut algorithm for particle-particle interaction.

3.1 Splitting by time discretization

Predictor

Given F^k, X^k and U^k.

$$X^{k+1}:=X^k+\tau U^k+\frac{\tau^2}{2\beta M}F^k,\quad U^{k+\frac12}:=U^k+\frac{\tau}{2\beta M}F^k. \qquad (16)$$

$$F^{k+1}=F(t^{k+1},X^{k+1}),\quad \breve U:=U^{k+\frac12}+\frac{\tau}{2\beta M}F^{k+1}. \qquad (17)$$

Combined Navier Stokes

Step 1 (Momentum equation)
Given u^k, u^{k-1}, p^k, χ^k, χ^{k-1}, $\breve{U}$, ω^k.
Set $u^\star = 2u^k - u^{k-1}$, $\omega^\star = 2\omega^k - \omega^{k-1}$.

Find $\mathfrak{u}^{k+1} \in H_c(\Omega_c)$ such that for all $\mathfrak{v} \in H_c(\Omega_c)$

$$\begin{aligned} & m(u^{k+1},v) && +\gamma k(u^\star;u^{k+1},v) + \gamma s(u^{k+1},v) + \\ & \frac{2}{3}\beta M U^{k+1}\cdot V && +\tfrac{2}{3}\beta I\omega^{k+1}\cdot\xi + \tfrac{\gamma}{2}\beta\,\omega^\star\times\left(I\omega^{k+1}\right)\cdot\xi = \\ & \gamma b(p^k,v) && +m(\tfrac{4}{3}u^k - \tfrac{1}{3}u^{k-1},v) + \gamma b(\tfrac{4}{3}\chi^k - \tfrac{1}{3}\chi^{k-1},v) + \\ & \frac{2}{3}\beta M\breve{U}\cdot V && +\tfrac{2}{3}\beta I\omega^{k}\cdot\xi - \tfrac{\gamma}{2}\beta\,\omega^k\times\left(I\omega^{k}\right)\cdot\xi. \end{aligned} \tag{18}$$

Step 2 (Computation of pressure correction)

Find $\chi^{k+1} \in H^1(\Omega_c)$ such that for all $\Psi \in H^1(\Omega_c)$

$$m(\nabla\chi^{k+1},\nabla\Psi) = \frac{1}{\gamma}b(\Psi,u^{k+1}). \tag{19}$$

Step 3 (Pressure update in rotational form)

Find $p^{k+1} \in L^2_0(\Omega_c)$ such that for all $q \in L^2_0(\Omega_c)$

$$m(p^{k+1},q) = m(p^k+\chi^{k+1},q) - b(q,\frac{2}{\mathrm{Re}}u^{k+1}). \tag{20}$$

Corrector

Given Θ^k, X^k, ω^k, ω^{k+1}, U^k and U^{k+1}.

$$\Theta^{k+1} = \left(\mathbb{I} - \frac{\tau}{2}\mathrm{R}\left[\omega^{k+1}\right]\right)^{-1}\left(\mathbb{I} + \frac{\tau}{2}\mathrm{R}\left[\omega^{k}\right]\right)\Theta^k. \tag{21}$$

$$X^{k+1} = X^k + \frac{\tau}{2}\left(U^k + U^{k+1}\right). \tag{22}$$

The above technique is used to to solve the highly coupled, highly nonlinear system of equations. The presented algorithm decouples the position and the orientation of the particles (X and Θ) from the combined Navier-Stokes equations (u, U, ω and p). These are then further decoupled by a pressure correction projection method. Thus the philosophy here is to finally split the complex system into a cascade of simple subproblems rather than using a (maybe more accurate but much more expensive) monolithic approach.

To be more precise, in order to discretize in time, the time interval $(0,T)$ is subdivided by discrete time instants: $0 = t^0 < t^1 < \cdots < t^N = T$. Denote by $\tau_{k+1} := t^{k+1} - t^k$. For simplicity a fixed time step size τ is used: $\tau_k = \tau$ for all $k = 1, \ldots N$. Moreover, define $\gamma = \frac{2}{3}\tau$.

Then in each time step Eqs. (12)–(15) are split into three substeps. The first is a *predictor* step for the new particle position and velocity, X^{k+1}, $\check{U}$, respectively. In the second step values for u^{k+1}, U^{k+1}, ω^{k+1} and p^{k+1} are computed based on a BDF2 scheme. The last step is a *corrector* for X^{k+1}, Θ^{k+1}.

The predictor step is a *Velocity Verlet* method, which is of second order [21, 15] and the common tool in particle dynamics.

The combined Navier Stokes equations are discretized by a projection method in *rotation form*, see [11, 10]. To this end, the time derivative $\partial_t u$ is replaced by a BDF2 scheme having good stability properties, while the equations for $\dot{U}$ and $\dot{\omega}$ are approximated by Crank-Nicolson differences, respectively.

The corrector uses the Crank-Nicolson scheme for time discretization.

3.2 Spatial discretization

The core problem in solving the time discretized system are the combined Navier Stokes equations and in particular Eq. (18). The crucial point in the spatial discretization is to define a discrete counterpart of $H_c(\Omega_c)$ and, moreover, the concrete realization of this non–standard finite element space. A brief description of how to solve this problem is given in the sequel, a more comprehensive presentation can be found in [17].

Let $\mathscr{T}$ be a triangulation of $\overline{\Omega}$. Since we are using the *Taylor–Hood* element for velocity and pressure, the basic finite element space for the velocity is given by the space of piecewise quadratic elements:

$$X(\Omega_c) = \left\{ (v,V,\xi) \,|\, v \in \left(C^0(\overline{\Omega}_c)\right)^2, v \in \left(P^k(T)\right)^2 \forall T \in \mathscr{T}, V \in \mathbb{R}^2, \xi \in \mathbb{R}, v = 0 \text{ on } \Gamma_D \right\}.$$

A discrete subspace of $H_c(\Omega_c)$ is now given by

$$X_c(\Omega_c) = \left\{ (v_c,V,\xi) \in X(\Omega_c) |\; v_c = V + \xi \times r \;\text{ in } P(t) \right\}.$$

For a given time step k the linear Eq. (18) may be rewritten with the bilinear form a, the corresponding operator $\mathscr{A}$, and the cumulative right hand side g: find $u \in X_c(\Omega_c)$ such that for all $v \in X_c(\Omega_c)$ it holds

$$a(u,v) =: (\mathscr{A}u, v) = (g,v). \tag{23}$$

To circumvent the explicit representation of $H_c(\Omega_c)$, a subspace projection $\pi : X \to X_c$ is used. With this operator (23) may be formulated in terms of the *standard* finite

element space $X(\Omega_c)$: find $\tilde{u} \in X(\Omega_c)$ such that for all $v \in X(\Omega_c)$ it holds

$$(\mathscr{A}\pi\tilde{u}, \pi v) = (g, \pi v). \tag{24}$$

Note that the solution u is now easily found by taking $u = \pi\tilde{u}$, where $\tilde{u}$ is a solution of Eq. (24). The above system now leads to the linear system of equations for the nodal vector $\tilde{U}$ of the form

$$\Pi^T A \Pi \tilde{U} = \Pi^T G, \tag{25}$$

where A is the system matrix corresponding to operator $\mathscr{A}$ and Π is a matrix representation of π. We call this method *subspace projection method*. Note that, when using iterative solvers, one can bypass to explicitly compute the modified system matrix $\Pi^T A \Pi$, but rather just needs to slightly modify the matrix vector product, because one only has to take into account the action of $\Pi^T A \Pi$ on a vector. Because the matrix Π is quite simple, its not necessary to store it explicitly. Instead, a short routine can perform the multiplication of Π and Π^T with a vector v. The following pseudo–code shows this computation.

```
! Multiplication (u,U,omega)=Pi*(v,V,xi)
subroutine Pmul(v,V,xi,u,U,omega)

   ! U, omega
   do ii=1,npart ! Number of particles
      U(:,ii)    = V(:,ii)
      omega(ii)  = xi(ii)
   end do

   ! u = rigid body motion in the particle
   do i=1,nk ! Number of DOFs
      if( isparticle(i) ) then
         ii= numpart(i)
         r(:)= x(:,i) - xpart(:,ii)
         u(1,i) = V(1,ii) - r(2)*xi(ii)
         u(2,i) = V(2,ii) + r(1)*xi(ii)
      else
         u(:,i)= v(:,i)
      end if
   end do

end subroutine
```

3.3 Adaptivity

One of the most important issues in simulating particulate flow is the numerical representation of the particle's geometry.

In Hu [12] a remeshing technique was used to explicitly follow the geometry in time, Wan and Turek [22] introduced a mesh deformation technique and Glowinski et al. [9] used Lagrange multipliers. In contrast to these methods, we use time dependent adaptively refined/coarsened grids based on the bisection method [1] to sufficiently resolve the region around the particle.

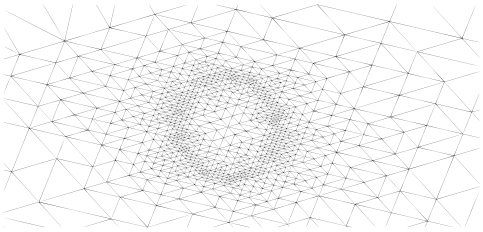


Fig. 2 Adaptive refined mesh around a particle. For an accurate representation it is useful to refine the mesh on the particle boundary.

The overall algorithm was implemented in the finite element flow solver NAVIER, for more details see [2].

3.4 Preconditioning

In general, the matrix $\Pi^T A \Pi$ (if the kernel would be factored out) is ill conditioned so that preconditioning is mandatory for an efficient solution strategy.

We developed a preconditioner based on inexact factorization that gave rather satisfying results, see [18].

3.5 Particle interaction

Efficient evaluation of the particle-particle interactions in Eq. (17) is crucial. For a large number of particles (more than, say, 1000) a naive implementation requiring $\mathcal{O}(n^2)$, n the number of dofs, would be prohibitive. Instead we use the Barnes-Hut algorithm, which reduces the complexity to $\mathcal{O}(n\log(n))$ with an acceptable loss of accuracy, see [3].

The idea of the algorithm is to merge the forces created by a group of neighboring particles into a single force of one pseudo-particle.

In addition to the long range Coulomb forces we also add short range repulsive forces in order to model particle collisions and avoid mutual penetration of particles. A similar approach is used for near particle-wall collision.

4 Computational examples

In this section we present some applications of the method described above. Quantitative validations can be found in [17]. Here we present further numerical experiments.

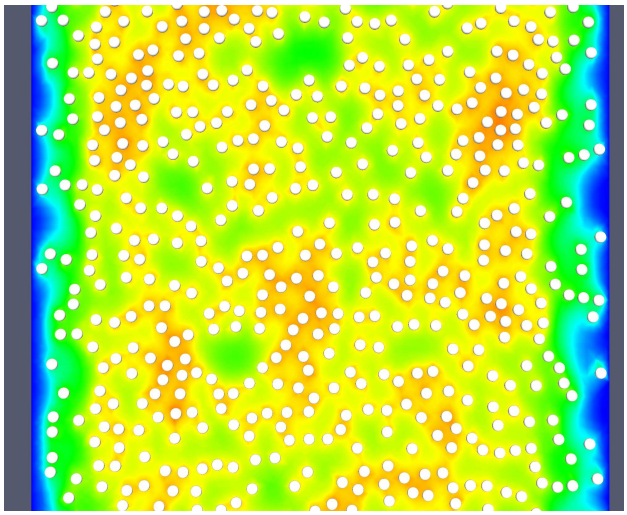


Fig. 3 Sedimentation of particles in 2D.

Fig. 3 shows a snapshot of a bunch of sedimenting particles (in 2D) under the influence of gravity.

The next experiment considers the sedimentation of two spherical particles in a cylindrical domaini in 3D. The particles are initially aligned on the center line, separated by a small distance of a few particle diameters in the starting configuration. When gravity starts acting one can observe the following situations.

- Both particles start accelerating. There is no interaction between them.
- “Drafting”: after a while the slipstream of the first particle causes the second one to accelerate a little more.
- “Kissing”: a near impact is inevitable as the second particle has a higher velocity than the first one. The slower particle is pushed by the faster one (the force is transferred by the viscous fluid).

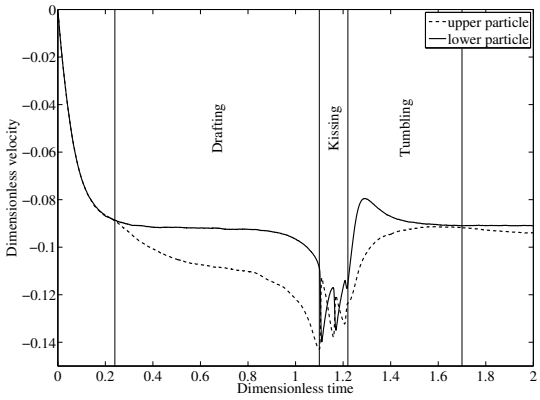


Fig. 4 Velocities of two particles traversing the phases of drafting – kissing – tumbling.

- "Tumbling": the above situation is unstable. To solve this conflict the slower particle moves aside, so that the faster particle can pass it. This can be interpreted as tumbling, when observed in relative coordinates.

These four phases described are displayed in Fig. 4.

The last example is a snapshot of the lid driven cavity in 3D with 1000 immersed particles, Fig. 5.

5 Discussion and Conclusion

A novel finite element method for the simulation of particulate flows was presented. Its key ingredients are: one domain approach, splitting in time, subspace projection method to account for the rigid body motion within the particles and time dependent adaptively refined meshes. The advantages of the method are its easy implementation and its efficiency. Only few modifications are needed to extend an existing Navier-Stokes code to simulate particulate flows by this method.

Acknowledgements This work was supported by the *Bayerische Forschungsstiftung* Grant 813-08, which is gratefully acknowledged.

References

1. Bänsch, E.: Local mesh refinement in 2 and 3 dimensions. Impact Comput. Sci. Engrg. **3**, 181–191 (1991)
2. Bänsch, E.: Simulation of instationary, incompressible flows. Acta Math. Univ. Comenianae **67**(1), 101–114 (1998)

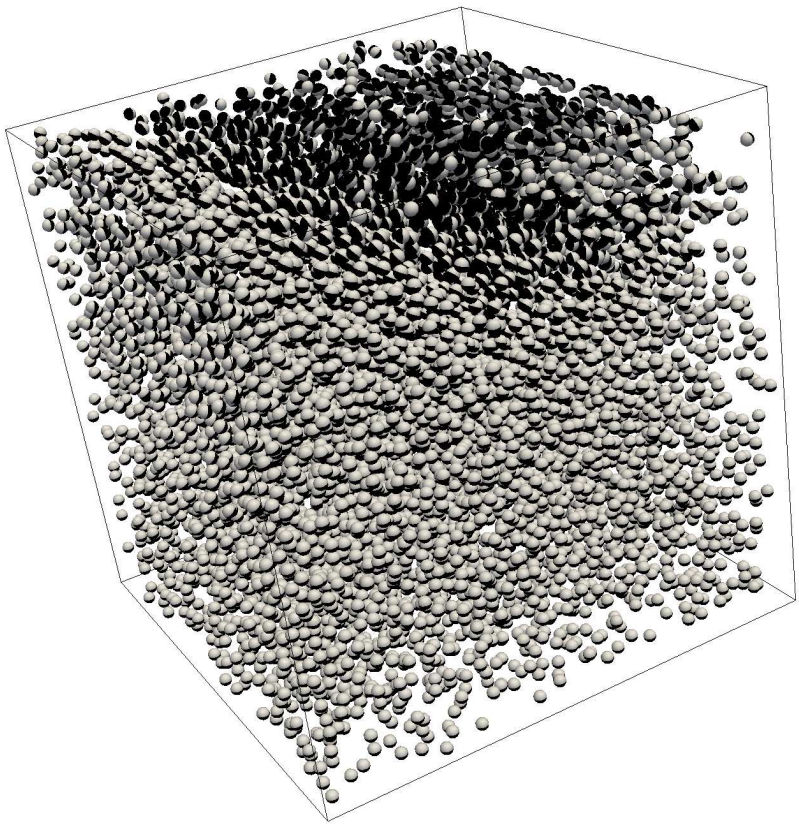


Fig. 5 Lid driven cavity 3D with 1000 particles.

3. Barnes, J., Hut, P.: A hierarchical $\mathcal{O}(n\log n)$ force-calculation algorithm. Nature **324**, 446–449 (1986)
4. Bönisch, S., Dunne, T., Rannacher, R.: Numerics of fluid-structure interaction. In: Hemodynamical flows, *Oberwolfach Semin.*, vol. 37, pp. 333–378. Birkhäuser, Basel (2008)
5. Bönisch, S., Heuveline, V.: On the numerical simulation of the instationary free fall of a solid in a fluid. I. The Newtonian case. Comput. Fluids **36(9)**, 1434–1445 (2007)
6. Esmaeeli, A., Tryggvason, G.: Direct numerical simulations of bubbly flows. Part 1: low Reynolds number arrays. J. Fluid Mech. **377**, 313–345 (1998)
7. Feng, J., Hu, H., Joseph, D.: Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 1. Sedimentation. J. Fluid Mech. **261**, 95–134 (1994)
8. Galdi, G.: Chapter 7: On the motion of a rigid body in a viscous liquid: A mathematical analysis with applications. pp. 653–791. North-Holland (2002)
9. Glowinski, R., Pan, T.W., Hesla, T., Joseph, D.: A distributed Lagrange multiplier/fictitious domain method for particulate flows. Int. J. Multiphase Flow **25**, 755–794 (1999)
10. Guermond, J.L., Minev, P., Shen, J.: An overview of projection methods for incompressible flows. Comput. Meth. Appl. Mech. Eng. **195**, 6011–6045 (2006)
11. Guermond, J.L., Shen, J.: On the error estimates for the rotational pressure-correction projection methods. Math. Comp. **73**, 1719–1737 (2004)

12. Hu, H.: Direct simulation of flows of solid-liquid mixtures. Int. J. Multiphase Flow **22**(2), 335–352 (1996)
13. Johnson, A., Tezduyar, T.: Simulation of multiple spheres falling in a liquid-filled tube. Comput. Methods Appl. Mech. Engrg. **134**, 351–373 (1996)
14. LeVeque, R., Li, Z.: Immersed interface methods for Stokes flow with elastic boundaries or surface tension. SIAM J. Sci. Comput. **18**, 709–735 (1997)
15. Martys, N., Mountain, R.: Velocity Verlet algorithm for dissipative-particle-dynamics-based models of suspensions. Phys. Rev. E **59**(3), 3733–3736 (1999)
16. Maury, B.: Direct simulation of 2d fluid-particle flows in biperiodic domains. J. Comput. Phys. **156**, 325–351 (1999)
17. Prignitz, R., Bänsch, E.: Simulation of particulate electrodynamc flows with the subspace projection method. Submitted
18. Reitsam, A.: Hierarchical preconditioning for P^TAP-systems. Diploma, FAU Erlangen-Nrnberg (2011)
19. Serre, D.: Chute libre dun solide dans un fluide visqueux incompressible. Existence. Jpn. J. Ind. Appl. Math. **4**(1), 99–110 (1987)
20. Veeramani, C., Minev, P., Nandakumar, K.: A fictitious domain formulation for flows with rigid particles: A non-Lagrange multiplier version. J. Comput. Phys. **224**(2), 867–879 (2007)
21. Verlet, L.: Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. Phys. Rev. **159**, 98–103 (1967)
22. Wan, D., Turek, S.: Fictitious boundary and moving mesh methods for the numerical simulation of rigid particulate flows. J. Comput. Phys. **222**, 28–56 (2007)

Optimized Schwarz waveform relaxation for nonlinear systems of parabolic type

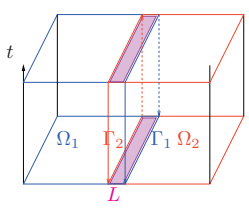
Florian Häberlein[1] and Laurence Halpern[1]

1 Schwarz waveform relaxation algorithms for a linear system

Let $\mathscr{L}$ be a partial differential operator, possibly acting on vector functions $(x,t) \mapsto u(x,t) \in R^d$, of the time variable t and the space variable $x = (x_1, x_2)$. The equation to be solved in $\Omega \times (0,T)$ is

$$\mathscr{L}u = F \text{ in } \Omega \times (0,T), \quad u(\cdot,0) = u_0 \text{ in } \Omega, \quad \mathscr{B}u = g \text{ on } \partial\Omega. \tag{1}$$

The domain Ω is split into subdomains Ω_i with possible overlap. Table 1 on the left shows the simplified case of a rectangle $\Omega = (A,B) \times (E,F)$ divided into two rectangles $\Omega_1 = (A, C+L) \times (E,F)$ and $\Omega_2 = (C,B) \times (E,F)$ with overlap L, this example will be the model case in the paper. On the right is described the alternate algorithm, *via* two *transmission operators* $\mathscr{B}_j$ on Γ_j. Boundary conditions are enforced on the other boundaries, of Dirichlet or Nemann type. A parallel Schwarz algorithm for elliptic equations was introduced by P.L. Lions in [14], extending the original Schwarz's domain decomposition algorithm for the Laplace equation in [16].



$$\begin{cases} \mathcal{L}u_1^k = F \text{ in } \Omega_1 \times (0,T) \\ u_1^k(\cdot,0) = u_0 \text{ in } \Omega_1, \ \mathcal{B}u_1^k = g \text{ on } \partial\Omega_1 \backslash \Gamma_1 \\ \mathcal{B}_1 u_1^k = \mathcal{B}_1 u_2^{k-1} \text{ on } \Gamma_1 \times (0,T) \end{cases}$$

$$\begin{cases} \mathcal{L}u_2^k = F \text{ in } \Omega_2 \times (0,T) \\ u_2^k(\cdot,0) = u_0 \text{ in } \Omega_2, \ \mathcal{B}u_2^k = g \text{ on } \partial\Omega_2 \backslash \Gamma_2 \\ \mathcal{B}_2 u_2^k = \mathcal{B}_2 u_1^{k-1} \text{ on } \Gamma_2 \times (0,T) \end{cases}$$

Table 1: Domain decomposition and Schwarz waveform relaxation algorithm

P.L. Lions also mentioned the possibility of using the algorithm for time dependent problem. However, it was recognized and analyzed as a waveform algorithm (see [13]) only in [7]. The authors defined the *Schwarz waveform relaxation algorithm*, which uses as transmission operators $\mathscr{B}_j \equiv Id$, corresponding to *Dirichlet* transmission conditions. The convergence was analyzed with various tools, such as maximum principle, Laplace transform in time. This algorithm enjoys superlinear convergence over small time intervals, linear convergence over large time intervals. A more detailed historical account can be found in [10]. On large time intervals, a Fourier analysis is useful. Considering a small overlap, the boundaries of the do-

[1] LAGA, Université Paris 13, 99 Avenue J-B Clément, 93430 Villetaneuse, France, e-mail: halpern@math.univ-paris13.fr

mains can be rejected to infinity, and Fourier transform in the second variable can be performed. This is the simplest way to proceed, but Fourier series on bounded intervals can be used as well, though the objects are heavier, see [4] for an example in structure mechanics. Numerical results show that the parameters obtained through the analysis in an infinite domain are relevant.

Consider for instance the advection-diffusion reaction problem, with

$$\mathscr{L}u := \partial_t u + \mathbf{a}\cdot\nabla u - \nu\Delta u. \tag{2}$$

The algorithm for the error e_j^k is the same, with vanishing data F and u_0. By Fourier transform in time and x_2, with dual variables τ and ξ, the Fourier transforms are explicitely given by

$$\hat{e}_1^k(x_1,\xi,\tau) = \eta_1^k(\xi,\tau)e^{\frac{L-x}{2\nu}(a-f(z))}, \quad \hat{e}_2^k(x_1,\xi,\tau) = \eta_2^k(\xi,\tau)e^{\frac{x}{2\nu}(a+f(z))},$$

with notations which will remain throughout the paper

$$z(\xi,\tau) = i(\tau + a_2\xi) + \nu\xi^2, \quad f(z) = \sqrt{a_1^2 + 4\nu z}.$$

The coefficients η_j^k are obtained recursively, using the transmission relations. They are governed by the convergence factor ρ_D, and given in the parallel case by

$$\rho_D(z,L) := e^{-\frac{L}{2\nu}f(z)}, \quad \eta_j^k = \rho_D(z,L)^k\,\eta_j^0.$$

ρ_D is identically equal to 1 when $L = 0$, so the algorithm is not convergent. For positive overlap, the high frequencies are damped exponentially. More precisely, for the rectangle case in Table 1, suppose the initial boundary value problem is solved by finite differences in time and space on a regular grid, with meshes Δt and $h = \Delta x_1 = \Delta x_2$. Suppose Dirichlet boundary conditions are enforced on $\partial\Omega$. Then the lowest frequency resolved by the grid on Γ_j is $\xi_m = \frac{\pi}{F-E}$, corresponding to a mode $\sin(\frac{\pi x_2}{F-E})$, while the highest frequency is $\xi_M = \frac{\pi}{h}$, corresponding to a mode $\sin(\frac{\pi x_2}{h})$. The highest and lowest frequencies in time are defined in the same way, by $\tau_m = \frac{\pi}{2T}, \tau_M = \frac{\pi}{\Delta t}$.

$$\tau_m = \frac{\pi}{2T}, \tau_M = \frac{\pi}{\Delta t}, \quad \xi_m = \frac{\pi}{F-E}, \xi_M = \frac{\pi}{h}, \quad K = z([\tau_m,\tau_M]\times[\xi_m,\xi_M]).$$

In this paper, we consider only implicit schemes, with Δt and h are comparable. Then the uniform convergence factor is given by

$$\sup_K |\rho_D(z,L)| \sim 1 - \frac{L}{2\nu}\,\mathrm{Re}\, f(\xi_m,\tau_m).$$

It tends linearly to 1 when the overlap tends to 0. For reasons of cost and memory, the overlap is usually a few mesh points only, which implies that the convergence

factor is highly dependent of the mesh size. It is therefore useful to design algorithms with a more robust convergence behavior.

Schwarz algorithms with Robin transmission conditions were proposed in [15], together with nonoverlapping subdomains. *Optimized Schwarz waveform relaxation algorithms* have afterwards been proposed, with or without overlap, to be able to accelerate the convergence of the algorithm. They use approximations of the Dirichlet-to Neumann operator, they are differential in time and in the boundary variable, and take here the form

$$\mathscr{B}_j u := (\mathbf{n}_j)_1(\nu\partial_1 u - \frac{a_1}{2}u) + \frac{p}{2}u + \frac{q}{2}(\partial_t u + a_2\partial_2 u - \nu\partial_{22}u). \tag{3}$$

When $q = 0$, the operators are called *Robin* operators, while for $q \neq 0$, they are referred to as *Ventcel* operators. The coefficients p and q are calculated such that they optimize the convergence factor of the algorithm in the Fourier variables. Define a first degree polynomial $s(z) = p + qz \in \mathbf{P}_1$. The choice of p and q is a particular case of the best approximation problem in the space $\mathbf{P}_n$ of complex polynomials with degree lower than n:

$$\rho(z,s,L) = \frac{s(z) - f(z)}{s(z) + f(z)}\, e^{-\frac{L}{2\nu} f(z)}, \quad |\rho(z^*,s^*,L)| = \inf_{s\in\mathbf{P}_n} \sup_{z\in K} |\rho(z,s,L)| := \delta_n^*(L). \tag{4}$$

The analysis of the best approximation problem for the advection-diffusion equation in one dimension in the Robin case ($n = 0$) has been made "by hand" in [6] for $\tau_m = 0$. Further general tools for well-posedness of the best approximation problem (4) have been set in [2] for the Robin case, and applied to the one-dimensional Ventcel-Schwarz algorithm. They are being extended in [1] to the 2-D case with a complete analysis and explicit asymptotic formulae. Well-posedness of the algorithm and convergence results, including variable coefficients and non planar boundaries in the nonoverlapping case, can be found in [11].

2 Optimized coefficients for the linear reactive transport system

We introduce a simplified system which has been used as a model in F. Häberlein's thesis on CO^2 sequestration. For the linearized system, we present optimized coefficients in closed form, extending previous results in [1]. A proof is given in the one-dimensional overlapping case, which is new. These coefficients will be used in the nonlinear case in §3.

Consider the system of equations for $\mathbf{u} = (u,v)$ in $\Omega \times (0,T)$,

$$\partial_t(\phi u) + \nabla\cdot(-\nu\nabla u + \mathbf{a}u) - R(u,v) = 0, \quad \partial_t(\phi v) + R(u,v) = 0, \tag{5}$$

where Ω is a bounded domain in $\mathbb{R}^d$ and u and v denote the concentration of the mobile and fixed species, respectively. $\phi > 0$ is the porosity which is supposed to be constant in time, $\nu \geq 0$ is the scalar diffusion-dispersion coefficient, $\mathbf{a} \in \mathbb{R}^d$ is

the Darcy velocity. All physical properties are supposed to be given and constant in time. $R(u,v)$ is a nonlinear function representing the chemical coupling term. The final goal is to be able to simulate general situations where the kinetic reaction rate is fully nonlinear. We present in §3 a test case with a semilinear model $R(u,v) = k(v - \Psi(u))$, where $k > 0$ represents the reactive surface and Ψ is a nonlinear function modeling an adsorption process, see Figure 3, left. The domain decomposition process relies on obtaining transmission conditions leading to a fast convergence of the iterative approach. Therefore we consider first a linear coupling term $R(u,v) = k(v - cu)$ where $k > 0$ represents the reactive surface and $c > 0$ an equilibrium constant. The linear case models a chemical reaction that reaches its equilibrium point at $v = cu$. By the same method of approximating the Dirichlet to Neumann map, Ventcell transmission conditions can be obtained:

$$\mathscr{B}_j \cdot \mathbf{u} := \pm(\nu\partial_1 - \frac{a_1}{2})\,u + \frac{p}{2}u + \frac{q}{2}(\phi\partial_t + a_2\partial_2 - \nu\partial_{22} + kc)\,u - \frac{q}{2}kv. \tag{6}$$

The convergence factor is still defined by (4), with z replaced by

$$Z(\xi,\tau,c) = z(\xi,\phi\tau) + kc\,\frac{i\phi\tau}{i\phi\tau + k}. \tag{7}$$

$Z(\xi,\tau,c)$ appears as a perturbation of the function $z(\xi,\phi\tau)$ introduced previously, and will be treated as a linear perturbation in the parameter c. The domain of optimization is $K(c) = Z([\tau_m,\tau_M]\times[\xi_m,\xi_M],c)$.

Warning: in this text, the proofs are based very often on asymptotic considerations. To alleviate the notations, we introduce the notation $Q \backsimeq h$ or $Q = \circ(h)$ if there exists $C \neq 0$ such that $Q \sim Ch$. The analysis below is an extension of that made in the case $c = 0$ described above. The formulas include the case $c = 0$. The important theoretical results in [2, 1] apply here, to give

1. Existence of solutions for the best approximation problem, overlap or not.
2. Uniqueness for small $L, \Delta t$ and h, in the Robin case $n = 0$.
3. Uniqueness for $L = 0$, small Δt and h, in the Ventcell case $n = 1$.
4. For $n = 0$ and 1, consider the real function

$$F(s,L) = \sup_{Z\in K(c)\cap\{\Re Z\geq 0\}} |\rho(Z,s(Z),L)|\,. \tag{8}$$

   If it has a local minimum in $\mathbf{P}_n$, it is the global minimum.

The last property will be decisive for the approximate computation of the best parameters.

Shortcuts are defined in one dimension by $f_m = f(Z(0,\tau_m,c))$, $f_M = f(Z(0,\tau_M,c))$.

Theorem 1. *For positive c, small $h \backsimeq \Delta t$, if $L = 0$ or $L \backsimeq h$, the best approximation problem* (4) *in $K(c)$ has a unique solution, whose coefficients are given in the 1-D case asymptotically in terms of $x_m = \Re(f(\tau_m))$, $x_M = \Re(f(\tau_M)) \sim \sqrt{\frac{2\nu\pi\phi}{\Delta t}}$:*

dimension	*method*	*overlap*	*parameters* (p^*, q^*)	$\delta^* \sim 1-2\frac{x_m}{p^*}$
1	*n=0, Robin*	$L=0$	$p_0^*(0)=\sqrt{\frac{x_m\lvert f_M\rvert^2-x_M\lvert f_m\rvert^2}{x_M-x_m}}$	$1-\propto(\Delta t^{\frac{1}{4}})$
1	*n=0, Robin*	$L>0$	$p_0^*(L)\sim p_0^*(0)$	$1-\propto(\Delta t^{\frac{1}{4}})$
1	*n=1, Ventcel*	$L\geq 0$	$p_1(L)^*\sim x_m^{\frac{3}{4}}x_M^{\frac{1}{4}}, \quad q_1^*(L)\sim\frac{2\nu p^*}{x_m x_M}$	$1-\propto(\Delta t^{\frac{1}{8}})$

In two dimensions, define , for $|a_2|\xi_m > \tau_m$, τ_0 *as the largest real root of*

$$\phi\tau\left(1+c\frac{k^2}{k^2+\tau^2\phi^2}\right)=|a_2|\xi_m,$$

the real function $g_1(s)=\dfrac{k^2}{s+k^2}$,

$$\xi_1=a_2\,\frac{(|\mathbf{a}|^2+4\nu kc(1-g_1(\phi\tau_m)))-\sqrt{(|\mathbf{a}|^2+4\nu kc(1-g_1(\phi\tau_m)))^2\dagger+16\nu^2(\phi\tau_m)^2(1+cg_1(\phi\tau_m))^2}}{8\nu^2\phi\tau_m(1+cg_1(\phi\tau_m))},$$

$$Z_w=\begin{cases} Z(\xi_1,\tau_m) & \text{if } \xi_m\leq|\xi_1|\leq\xi_M,\\ Z(\tau_0,-sign(a_2)\xi_m) & \text{if } |\xi_1|\leq\xi_m \text{ and } \Re f(\tau_0,-sign(a_2)\xi_m)\leq\Re f(\tau_m,-sign(a_2)\xi_m),\\ Z(\tau_m,-sign(a_2)\xi_m) & \text{otherwise.}\end{cases}$$

$x_w=\Re Z_w$.

The best coefficients for the Robin-Schwarz algorithm ($n=0$) *are*

overlap	*parameter* p^*	$\delta^*\sim 1-2\frac{x_w}{p^*}$
$L=0$	$p_0^*(0)\sim\sqrt{\frac{2\nu\pi x_w\phi}{\Delta t}}$	$1-\propto(\Delta t^{\frac{1}{2}})$
$L>0$	$p_0^*(L)\sim\sqrt[3]{\frac{\nu x_w^2}{2L}}$	$1-\propto(L^{\frac{1}{3}})$

Define the function

$$g(t)=\frac{2t-\sqrt{t^2+1}}{t^2+1},$$

and for $Q<Q_0\approx 0.36900$, $t_2(Q)$ *is the only root of* $g(t)=Q$ *larger than* $t_0=\sqrt{54+6\sqrt{33}}/6\approx 1.567618292$,

$$P(Q)=\begin{cases}\sqrt{1+\sqrt{t_2(Q)^2+1}}\left(\frac{1}{\sqrt{t_2(Q)^2+1}}+Q\right) & \text{if } Q<Q_1\approx 0.1735,\\ 1+Q & \text{if } Q>Q_1.\end{cases}\tag{9}$$

Defining $C=\Delta t/h$, *the best coefficients for the Ventcel-Schwarz algorithm* ($n=1$) *are*

overlap	p_1^*	q_1^*	$\delta^*\sim 1-2\frac{x_w}{p_1^*}$
$L=0$	$p_1^*(0)\sim\begin{cases}\sqrt[4]{\frac{\nu x_w^3\pi}{h}} & \text{if } Cx_w<2,\\ \sqrt[4]{\frac{8\nu x_w\pi}{hC\left(P(\frac{2}{Cx_w})\right)^2}} & \text{if } Cx_w>2,\end{cases}$	$q_1^*(0)\sim\frac{2p_1^*(0)\pi}{hx_w}$	$1-\propto(h^{\frac{1}{4}})$
$L>0$	$p_1^*(L)\sim\sqrt[5]{\frac{\nu x_w^4}{2L}}$	$q_1^*(L)\sim\frac{2\nu x_w^2}{p_1^*(L)^3}$	$1-\propto(L^{\frac{1}{5}})$

Proof. It relies on the use of the explicit formulations in [1] for $c = 0$, together with a continuation argument. We present in detail the analysis for the Robin transmission condition with overlap. Define

$$R_0(\tau, s) = \left| \frac{s - f(Z)}{s + f(Z)} \right|^2, \quad R(\tau, s, L) = R_0(\tau, s) e^{-L\Re f(Z)/\nu}. \tag{10}$$

Lemma 1. *In one dimension, for $\tau_M \gg 1$ and $L \ll 1$ with $L \backsimeq \tau_M^{-\lambda}$, the minmax problem* (4) *in $K(c)$ with $n = 0$ has a unique solution $(s_0^*(L), \delta_0^*(L))$.*

- *If $0 < \lambda < \frac{3}{4}$, it is the unique solution of the equation*

$$R(\tau_m, s, L) = R(\tau_+, s, L), \tag{11}$$

where $\tau_+(s, L)$ is the unique local maximum point of $R(\cdot, s, L)$. It is asymptotically given by

$$s_0^*(L) \sim \sqrt[3]{\frac{(\Re(f(\tau_m)))^2 L}{2\nu}} \quad \delta_0^*(L) \sim 1 - 2\sqrt[3]{\frac{\Re(f(\tau_m)) L}{2\nu}}, \tag{12}$$

- *If $\frac{3}{4} < \lambda \leq 1$, it is the unique solution of the equation*

$$R(\tau_m, s, L) = R(\tau_M, s, L). \tag{13}$$

It is asymptotically given by

$$s_0^*(L) \sim s_0^*, \quad \delta_0^*(L) \sim \delta_0^*. \tag{14}$$

Remark 1. Note that if λ is close to 0, then $\delta_0^*(L) = 1 - \propto(\sqrt[3]{L})$, which gives the best behavior, independent of Δt. For the Dirichlet case, we would have

$$\sup_K |\rho_D(\tau, L)| = 1 - \propto(L).$$

If $\lambda = 1$, which is the case if the overlap contains a few grid points, then the overlap does not improve the convergence. We will see that it is not the case in higher dimension.

Proof of the Lemma Introduce the curve $\mathscr{F} : \tau \in \mathbb{R}^+ \mapsto f(\tau) \in \mathbb{C}$. The domain $K(c)$ is $\mathscr{F}([\tau_m, \tau_M])$. The proof has four steps.

1. Study the graph of $\mathscr{F}$, see Figure 1.
2. Existence and uniqueness of a minmax reached at $(s_0^*(L), \delta_0^*(L))$ follows from the theoretical results above.
3. Prove that if L is small, s is large, and Ls is small, the function $\tau \mapsto R(\tau, s, L)$ has a unique stationary point $\tau_+ \sim s/L\phi$ corresponding to a maximum.
4. Prove that for small L, there is a unique $\overline{s_0^*}(L)$ such that $R(\tau_m, s, L) = R(\tau_+, s, L)$ or $R(\tau_M, s, L)$, and that it satisfies the assumptions in the previous item.
5. Prove that $\overline{s_0^*}(L)$ is a strict real minimum point of $F(\cdot, L)$.

6. Conclude by theoretical results that $\overline{s_0^*}(L) = s_0^*(L)$.

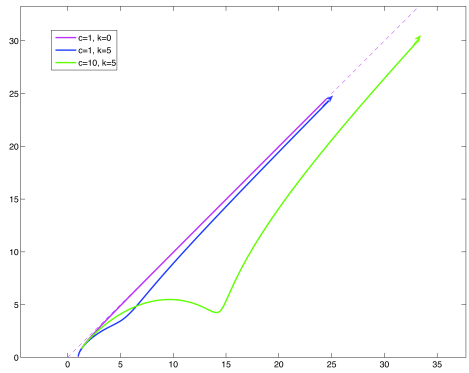


Fig. 1 Geometric representation of the function $\mathscr{F}$ defining $K(c)$, for $c = 1$, $k = 0$ (magenta), $c = 1$, $k = 5$ (blue), $c = 10$, $k = 5$ (green). The direction of increasing τ is indicated by the arrow.

The real and imaginary parts of f, $x(\tau)$ and $y(\tau)$, are defined by:

$$\begin{cases} x^2 - y^2 = a_1^2 + 4\nu \dfrac{kc\tau^2\phi^2}{k^2+\tau^2\phi^2}, \\ 2xy = 4\nu\tau\phi\left(1 + \frac{k^2 c}{k^2+\tau^2\phi^2}\right), \\ x \geq x_m > 0, \quad y \geq 0. \end{cases} \tag{15}$$

In the (x,y) plane, the curve $\mathscr{F}$ lies between the real axis and the bisectrix $(y = x)$. For further investigations, the derivatives of x and y are needed. To simplify the notations, introduce

$$\omega = \phi\tau, \quad g_1(s) = \frac{k^2}{s+k^2}, \quad g_2(s) = 1 - cg_1(s) + 2cg_1(s)^2,$$

and differentiate (15) to obtain the derivatives of x and y, in terms of x, y, g_1, and g_2 as:

$$\begin{cases} x^2 - y^2 = a_1^2 + 4\nu kc(1 - g_1(\omega^2)) \\ 2xy = 4\nu\omega(1 + cg_1(\omega^2)) \end{cases}, \quad \begin{pmatrix} \partial_\tau x \\ \partial_\tau y \end{pmatrix} = \frac{2\nu\phi}{x^2+y^2} \begin{pmatrix} \frac{2c}{k}\omega g_1^2(\omega^2)x + g_2(\omega^2)y \\ -\frac{2c}{k}\omega g_1^2(\omega^2)y + g_2(\omega^2)x \end{pmatrix}. \tag{16}$$

The zeros of $\partial_\tau x$ exist only at points τ with $g_2(\omega^2) < 0$, which happens only if $c > 8$ and $g_1(\omega^2) \in]\tilde{g}_1^1, \tilde{g}_1^2[\subset]0,1[$, with $\tilde{g}_1^1 = \frac{c-\sqrt{c^2-8c}}{4c}$ and $\tilde{g}_1^2 = \frac{c+\sqrt{c^2-8c}}{4c}$. Accordingly $\partial_\tau y$ vanishes only at points τ with $g_2(\omega^2) > 0$, which happens if $c > 8$ and $g_1(\omega^2) \neq \in]\tilde{g}_1^1, \tilde{g}_1^2[$, or $c < 8$.

To solve $\partial_\tau x = 0$, it will be easier to rewrite it in terms of $g_1(\omega^2) < 0$ only. To do so, multiply the equation $\partial_\tau x = 0$ successively by x and by y, then replace $xy = 2\nu\omega(1+cg_1)$. In the resulting equation replace $\omega^2 g_1(\omega^2) = k^2(1-g_1(\omega^2))$, and finally insert these values into the equation $x^2 - y^2 = a_1^2 + 4\nu kc(1-g_1(\omega^2))$, to obtain that $\partial_\tau x(\tau) = 0$ is equivalent to

$$g_1(\omega^2) \text{ is a root of } Q \text{ in } (\tilde{g}_1^1, \tilde{g}_1^2), \text{ with}$$
$$Q(X) = -4c^2(c+2b+2)X^4 + c^2(3c+4b+8)X^3 - c(3c+4b+4)X^2 + cX - 1.$$

Computing the derivatives of Q, it is easy to see that Q has a maximum point in $(0,1)$. Since Q has alternate coefficients, it cannot have negative zeros. Compute $Q(0) = -1$, $Q(1) < 0$. $Q(\tilde{g}_1^j) = 4c^2(\tilde{g}_1^j)^3(1-\tilde{g}_1^j)(1+c\tilde{g}_1^j) > 0$. This proves that Q has two roots in $(0,1)$, outside $(\tilde{g}_1^1, \tilde{g}_1^2)$, which indeed correspond to zeros of $\partial_\tau y$. This implies that x is an increasing function of τ, y' vanishes for two values of τ, and the curve has the behavior depicted in Figure 1.

2. Rewrite the convergence factor R with $L = 2\nu\ell$ as

$$R_0(\tau, s) = \frac{(x-s)^2 + y^2}{(x+s)^2 + y^2}, \quad R(\tau, s, L) = R_0(\tau, s)e^{-2\ell x}$$

Compute for fixed s the derivative of R with respect to τ.

$$\partial_\tau R(\tau, p, L) = (\partial_\tau R_0(\tau, s) - 2\ell\partial_\tau x R_0(\tau, s))e^{-2\ell x} = \frac{2\nu\phi S(\tau, p, \ell)}{|f|^2|f+p|^4}$$

with

$$S(\tau, s, \ell, c) = \left(4s(x^2 - y^2 - s^2) - 2\ell|f^2 - s^2|^2\right)\left(\frac{2c}{k}\omega g_1^2 x + g_2 y\right) + 8sxy\left(-\frac{2c}{k}\omega g_1^2 y + g_2 x\right).$$

Suppose ℓ small, s large, and ℓs small. For $c = 0$, S is a bi-quadratic polynomial in the x variable

$$\tilde{S}(x, s, \ell) = -4\ell x^4 + 4(\ell b^2 + s)x^2 - \ell b^2(b^2 - 2s^2) + 2s(b^2 - s^2).$$

$\tilde{S}$ has two positive roots, which behave asymptotically as $x_- \sim s$ and $x_+ \sim \sqrt{s/\ell}$, corresponding to two values of τ, $\tau_- \sim \frac{s^2}{2\nu\phi} \ll \tau_+ \sim \frac{s}{2\nu\ell\phi}$. Since R tends to 0 at infinity, τ_- corresponds to a minimum, and τ_+ to a maximum of R.

We now extend the solution to positive c. A careful computation shows that

$$\partial_c S(\tau_\pm, s, \ell, c) \sim 16s\nu x_\pm \neq 0.$$

Therefore, by the implicit function theorem, in a neighborhood of 0, $0 \le c \le c_0$, the root τ_- (resp. τ_+) continues in a minimum point $\tau_-(c)$, (resp. maximum point $\tau_+(c)$) with $\tau_\pm(0) = \tau_\pm$. They have the same asymptotic behavior $\tau_+(c) \sim s/2\nu\ell$ (resp. $\tau_-(c) \sim s^2/2\nu$) independent of c, and one can iterate the argument, showing for any c the existence of a function $\tau_+(c) \sim \frac{s}{2\nu\ell\phi}$ (resp. $\tau_-(c) \sim \frac{s^2}{2\nu}$) with

$S(\tau_\pm(c),s,\ell,c)=0$. They remain indeed global maximal and minimal points: suppose that there exists another root τ of S, and examine its asymptotic behavior. Since $\partial_\tau x(\tau)>0$, it can not be at finite distance, since then we would have $S(\tau,p,\ell,c)\sim -4s^3x'<0$. Suppose now that $\tau \backsimeq \ell^{-\theta}$ with $\theta>0$. Then the principal part of S is:

$$-4\ell(x(\tau))^4+4p(x(\tau))^2-p^3(p\ell+2)$$

whose roots are equivalent to those of S, proving that there is no other extremal point than $\tau_\pm(c)$. Then

$$\sup_{\tau\in K} R(\tau,s,L)=\begin{cases}\max(R(\tau_m,s,L),R(\tau_+,s,L)) & \text{if } \tau_+<\tau_M,\\ \max(R(\tau_m,s,L),R(\tau_M,s,L)) & \text{if } \tau_+>\tau_M,\end{cases}$$

3. Compute now $\partial_s R(\tau,s,L)=(\partial_s R(\tau,s,0))e^{-\ell x}$. It is easy to see that $R(\tau_m,s,L)$ is an increasing function of s, $R(\tau_+,s,L)$ a decreasing function of s, and $R(\tau_M,s,L)$ has a minimum reached for $s=|f(\tau_M)|$.

If $\lambda<\frac{3}{4}$, asymptotic considerations show that there exists a $\bar{s}_0^*$ such that $R(\tau_m,s,L)-R(\tau_+,s,L)$=0, and that

$$\sup_{\tau\in K} R(\tau,s,L)=\begin{cases}R(\tau_+,s,L) & \text{for } s<\bar{s}_0^*,\\ R(\tau_m,s,L) & \text{for } s>\bar{s}_0^*.\end{cases}$$

The other case is similar.

4. To prove that it is a strict local minimum, proceed as in [1] and evaluate asymptotically the sign of $\partial_p R(\tau_+,\bar{s}_0^*(L),L)\times\partial_p R(\tau_m,\bar{s}_0^*(L),L)<0$.

2.1 Performances of different transmission conditions

In this test case in $\Omega=(0,1)\times(0,1)$, the diffusion parameter is $\nu=1$, advection is $\mathbf{a}=(1\cdot10^{-2},5\cdot10^{-2})$, the reactivity coefficient is set to $k=5$ with an equilibrium parameter of $c=10$. The finite volumes method is described in [8]. The discretization parameters are $\Delta t=\Delta x=\Delta y=2\cdot10^{-2}$. The domain Ω is split into $\Omega_1=[0,0.5+L]\times[0,1]$ and $\Omega_2=[0.5,1]\times[0,1]$. A minimal overlap of size $L=\Delta x$ is used. A random initial guess is imposed on the interface Γ_1. The results are plotted in Figure 2. The expected behavior takes place. The best convergence behaviour is obtained with optimised Ventcel conditions with overlap which reach the error precision of 10^{-10} in only 6 iterations.

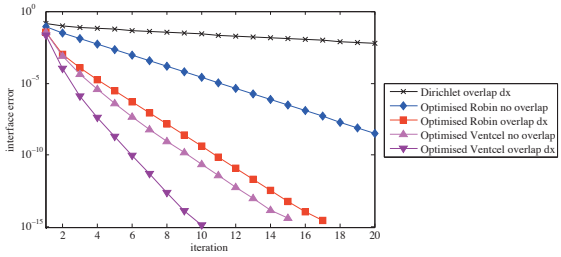


Fig. 2 Iterations versus error of the domain decomposition iterates

3 Newton-Schwarz waveform relaxation for the nonlinear system

The Schwarz waveform relaxation algorithm was used for the semilinear heat equation $\partial_t u - c^2(x)\partial_{xx}u + f(u) = 0$ in [5] . Under the condition that $f'(x) \leq a$, the same convergence behavior as in the linear case was exhibited and analyzed. Optimized Schwarz waveform relaxation algorihtm, with nonlinear transmission conditions were first introduced in [11], for the semilinear wave equation. In [3], the semilinear advection-diffusion reaction equation in 2 dimensions was considered, $\partial_t u - \nu\Delta u + f(u) = 0$, where f is constrained only to be in $C^2(\mathbb{R})$, with $f(0) = 0$. Nonoverlapping Robin-Schwarz and Ventcell-Schwarz where proposed and analyzed. The main difficulty in this case is that each iterate in Table 1 is solution of a nonlinear problem, whose solution has to be defined properly, and has its own time of existence T_j^n. The sequence $(T_j^n)_n$ is decreasing, and it must be shown that there is a lower bound T_* for these times. Then the convergence is achieved inside $(0, T_*)$. From a numerical point of view, a nonlinear system has to be solved in each subdomain at every step, which has been implemented with $\mathbf{P}_1$ finite elements in space, and a linearly implicit Euler scheme in time. It turns out that the requirement of small time interval given by the existence analysis is not compelling (see also [11]). Furthermore nonlinear transmission condition where the coefficients p and q depend on the iterates through the formulas of §1 were successfully implemented.

For the nonlinear reactive transport system, with suitable assumptions on the coefficients, the same methods apply, for the existence and convergence analysis. However, acceleration must be obtained. This has been done in F. Häberlein's thesis [8], where several scenarii where studied. First, writing the Schwarz iteration in an interface substructuring manner, it is seen as a fixed point iteration for the interface problem, preconditioned by the domain decomposition with transmission conditions given by the $\mathscr{B}_j$. It will be called *Classical approach*. For steady elliptic problems, the resolution of the interface problem is accelerated by a Krylov algorithm (see [17]). In this time-dependent non-linear frame, it is treated by a Newton-Krylov algorithm (called *Nested Iteration Approach*). Each iteration requires the resolution of smaller time-dependent nonlinear systems in the subdomains. This approach has been successfully implemented and described in [9]. An interesting other approach is called *Common iteration approach*. It is a Newton-Schwarz Krylov approach (see

[12]) with the Jacobian explicitely computed.

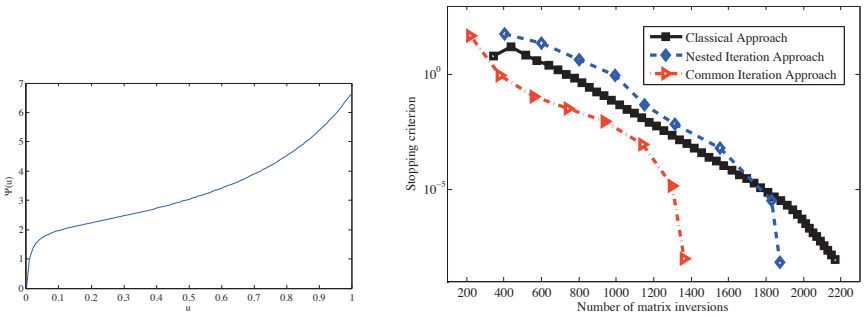
$$U^{k+1} = U^k + h, \quad \partial_t h - \nu \Delta h + f'(U^k)h = -(\partial_t U^k - \nu \Delta U^k + f(U^k)).$$

The linear problem above is solved by an optimized Ventcell-Schwarz domain decomposition algorithm, accelerated by Krylov. The approach requires in every iteration of the outer loop (indices in n) to set up a right hand side-vector that demands to solve two linear problems in the subdomains. Moreover, in the matrix-vector multiplication inside the Krylov-method, only linear problems in the subdomains are evaluated. No nested nonlinear iterative method is needed. For this reason and in contrast to the approach above, this approach was called 'Common Iteration Approach' (CIA) due to the common iterative approach of the nonlinear character of the monodomain problem. The name "Newton-Schwarz-Krylov" can be used in order to explain the order of application of the different numerical tools: The global problem is first attacked by a Newton-type method. At every iteration, the resulting linear problem is decomposed by a Schwarz-type algorithm where the problem is reduced to the interface variables. The resulting linear system is then solved by a Krylov-type method.

The next simulation shows nonoverlapping Robin-Schwarz simulations in domain $\Omega = [0,1] \times [0,1] \subset \mathbb{R}^2$ with the subdomains $\Omega_1 = [0,0.5] \times [0,1]$ and $\Omega_2 = [0.5,1] \times [0,1]$. The considered time window is $t \in [0,1]$. Physical parameters are $\phi = 1$, $\nu = 1.5$, $\mathbf{a} = (5 \cdot 10^{-2}, 1 \cdot 10^{-3})$. The nonlinear coupling term is defined by $R(u,v) = k(v - \Psi(u))$ where

$$\Psi(u) = \frac{Q_s K_L u}{(1 + K_L u - K_S u)(1 - K_S u)}$$

is the BET isotherm law with $k = 100$, $Q_S = 2$, $K_S = 0.7$ and $K_L = 100$ (cf. figure 3, left). BET theory is a rule for the physical adsorption of gas molecules on a solid surface and serves as the basis for an important analysis technique for the measurement of the specific surface area of a material. One observes the quadratic convergence



BET Isotherm Convergence history

Fig. 3 Nonlinear simulation with 200 points per space dimension

of the new approaches since they are Newton-based, the quadratic convergence is observed late in the history since the initial guess (randomly chosen) is far from the exact solution. The classical approach shows only a superlinear convergence, also in this case, the superlinear character appears late in the convergence history.

References

1. Bennequin, D., Gander, M.J., Gouarin, L., Halpern, L.: Optimized Schwarz waveform relaxation for advection reaction diffusion equations in two dimensions. In preparation (2013)
2. Bennequin, D., Gander, M.J., Halpern, L.: A homographic best approximation problem with application to optimized Schwarz waveform relaxation. Math. of Comp. **78**(265), 185–232 (2009)
3. Caetano, F., Halpern, L., Gander, M., Szeftel, J.: Schwarz waveform relaxation algorithms for semilinear reaction-diffusion. Networks and heterogeneous media **5**(3), 487–505 (2010)
4. Cissé, I.: Décomposition de domaines pour des structures hétérogènes. Ph.D. thesis, Université Paris 13 (2011)
5. Gander, M.J.: A waveform relaxation algorithm with overlapping splitting for reaction diffusion equations. Numer. Linear Alg. Appl **6**, 125–145 (1998)
6. Gander, M.J., Halpern, L.: Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems. SIAM J. Numer. Anal. **45**(2), 666–697 (2007)
7. Gander, M.J., Stuart, A.M.: Space time continuous analysis of waveform relaxation for the heat equation. SIAM J. Sci. Comput. **19**, 2014–2031 (1998)
8. Häberlein, F.: Time-space domain decomposition methods for reactive transport applied to CO_2 geological storage. PhD Thesis, University Paris 13 (2011)
9. Häberlein, F., Halpern, L., Michel, A.: Schwarz waveform relaxation and Krylov accelerators for nonlinear reactive transport. In: R.E. Bank, M.J. Holst, O.B. Widlund, J. Xu (eds.) Domain Decomposition Methods in Science and Engineering XX, pp. 409–416. Springer (2013)
10. Halpern, L.: Optimized Schwarz waveform relaxation: roots, blossoms and fruits. In: Domain Decomposition Methods in Science and Engineering XVIII, *Lect. Notes Comput. Sci. Eng.*, vol. 70, pp. 225–232. Springer (2009)
11. Halpern, L., Japhet, C., Szeftel, J.: Optimized Schwarz waveform relaxation and discontinuous Galerkin time stepping for heterogeneous problems. SIAM J. on Numer. Anal. **50**(5), 2588–2611 (2012)
12. Knoll, D., Keyes, D.: Jacobian-free Newton–Krylov methods: a survey of approaches and applications. J. of Comp. Phys. **193**, 357–397 (2004)
13. Lelarasmee, E., Ruehli, A.E., Sangiovanni-Vincentelli, A.L.: The waveform relaxation method for time-domain analysis of large scale integrated circuits. IEEE Trans. on CAD of IC and Syst. **1**, 131–145 (1982)
14. Lions, P.L.: On the Schwarz alternating method. I. In: R. Glowinski, G.H. Golub, G.A. Meurant, J. Périaux (eds.) First International Symposium on Domain Decomposition Methods for Partial Differential Equations, pp. 1–42. SIAM, Philadelphia, PA (1988)
15. Lions, P.L.: On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In: T.F. Chan, R. Glowinski, J. Périaux, O. Widlund (eds.) Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989. SIAM, Philadelphia, PA (1990)
16. Schwarz, H.A.: Über einen Grenzübergang durch alternierendes Verfahren. Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich **15**, 272–286 (1870)
17. Toselli, A., Widlund, O.: Domain Decomposition Methods - Algorithms and Theory, *Springer Series in Computational Mathematics*, vol. 34. Springer (2005)
18. Tran, M.B.: Overlapping domain decomposition: Convergence proofs. In: R.E. Bank, M.J. Holst, O.B. Widlund, J. Xu (eds.) Domain Decomposition Methods in Science and Engineering XX, *Lect. Notes Comput. Sci. Eng.*, vol. 91, pp. 519–526. Springer (2013)

Domain Decomposition for Boundary Integral Equations via Local Multi-Trace Formulations

Ralf Hiptmair[1], Carlos Jerez-Hanckes[2], Jin-Fa Lee[3], and Zhen Peng[4]

1 Introduction

This article is devoted to a formal derivation and discussion of a class of boundary integral equation (BIE) formulations that have recently been introduced for second-order transmission problems. We chose to dub this class "local multi-trace BIE formulations" (MTF), which is inspired by two key features of its members:

(i) The methods rely on at least two pairs of trace data as unknowns on interfaces. The accounts for the attribute "multi-trace".
(ii) Formally, they are constructed by taking into account transmission conditions pointwise or, at least, on parts of sub-domain boundaries, which is indicated by the "local" attribute.

Initially, the development of these new methods was pursued independently by numerical analysts and in computational electrical engineering, driven by different objectives. In numerical analysis, the focus was on composite structures, that is, partial differential equations with piecewise constant coefficients. There, the main motivation was to find first-kind boundary integral formulations that, after Galerkin boundary element discretization, are amenable to operator preconditioning, a possibility not offered by classical approaches, see [3, Section 4]. In engineering, researchers were guided by a domain decomposition paradigm, aiming to localize boundary integral equations for electromagnetic wave propagation at artificial interfaces for the sake of parallelization and block-preconditioning.

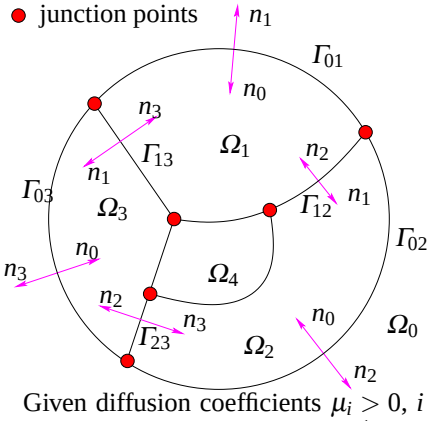
Both research efforts have been fairly successful: on the one hand, a comprehensive theoretical understanding of the simplest representative of a local multi-trace BIE formulations for Helmholtz transmission problem could be achieved in [8]. In a wider context the method is also covered in [3]. On the other hand, a host of impressive applications of multi-trace methods is documented in computational electromagnetism. A surface integral equation domain decomposition method based on multi-trace formulation is presented in [15, 14] for time-harmonic electromagnetic wave scatterings from homogeneous targets. The treatment of general bounded composite targets is discussed in [13].

This article looks at MTF from a mathematical point of view, but, inspired by the developments in the engineering community, adopts a different and more general

[1] Seminar for Applied Mathematics, ETH Zurich, CH-8092 Zürich, Switzerland, e-mail: hiptmair@sam.math.ethz.ch ·[2] Escuela de Ingeniería, Pontificia Universidad Católica de Chile, Santiago, Chile, e-mail: cjerez@ing.puc.cl ·[3] ElectroScience Laboratory, The Ohio State University, Columbus, OH, USA, e-mail: lee.1863@osu.edu ·[4] University of New Mexico, Albuquerque, NM, USA, e-mail: zpeng@ece.umn.edu

perspective compared to [8]. This work is mainly conceptual and does not aim to pursue any comprehensive analysis. Rather it is meant to chart new ideas and directions of research. We have not included any numerical results nor are we going to discuss details of Galerkin discretization by means of boundary elements. Detailed studies of convergence of multi-trace BIE for 2D acoustic scattering discretized by means of low-order boundary elements (BEM) are reported in [8, Sect. 5]. Concerning the application of multi-trace methods for solving electromagnetic scattering problems, convergence studies can be found in [13] for scattering at both single homogeneous objects and composite penetrable objects. Several complex large-scale simulations are covered in [14] and demonstrate the capability of these methods to model multi-scale electrically large targets.

2 Transmission Problems



Let $\Omega_i \subset \mathbb{R}^d$, $d = 2,3$, $i = 0,\ldots,N$, be disjoint open connected Lipschitz "material sub-domains" that form a partition in the sense that $\mathbb{R}^3 = \overline{\Omega}_0 \cup \cdots \cup \overline{\Omega}_N$. Among them only Ω_0 is unbounded. Two adjacent sub-domains Ω_i and Ω_j are separated by their common interface Γ_{ij}, whose union forms the skeleton Σ. For $N > 1$ the skeleton Σ will usually not be orientable, nor be a manifold.

Given diffusion coefficients $\mu_i > 0$, $i = 0,\ldots,N$, we focus on the model transmission problem that seeks $U_i \in H^1_{\mathrm{loc}}(\Omega_i)$, $i = 0,\ldots,N$, solving

$$\mathsf{L}_i U_i := -\operatorname{div}(\mu_i \mathbf{grad} U_i) + U_i = 0 \quad \text{in } \Omega_i \,, \tag{1a}$$

$$U_i|_{\Gamma_{ij}} - U_j|_{\Gamma_{ij}} = 0 \,, \quad \mu_i \frac{\partial U_i}{\partial n_i}\bigg|_{\Gamma_{ij}} + \mu_j \frac{\partial U_j}{\partial n_j}\bigg|_{\Gamma_{ij}} = 0 \quad \text{on } \Gamma_{ij} \,, \tag{1b}$$

plus suitable decay conditions at infinity for $U - U_{\mathrm{inc}}$, where the "incident field" U_{inc} is an entire solution of $\mathsf{L}_0 U_{\mathrm{inc}} = 0$ on Ω_0 [11, Ch. 8]. The weak formulation of (1) is posed on the Sobolev space $H^1(\mathbb{R}^3)$.

The transmission conditions (1b) connect two kinds of canonical traces on both sides of interfaces. These traces are the Dirichlet trace $\mathsf{T}_{D,i}$, and Neumann trace $\mathsf{T}_{N,i}$, defined for smooth functions on $\overline{\Omega}_i$ through

$$\mathsf{T}_{D,i} U_i := U_i|_{\partial\Omega_i} \quad , \quad \mathsf{T}_{N,i} U_i := \mu_i \mathbf{grad} U_i \cdot n_i|_{\partial\Omega_i} \,. \tag{2}$$

They can be extended to continuous operators [16, Sect. 2.6 & 2.7] [1]

$$\mathsf{T}_{D,i} : H^1(\Omega_i) \to H^{\frac{1}{2}}(\partial\Omega_i) \quad , \quad \mathsf{T}_{N,i} : H(\Delta,\Omega_i) \to H^{-\frac{1}{2}}(\partial\Omega_i) \, . \tag{3}$$

Then, (1b) can be recast as

$$\begin{pmatrix} \mathsf{T}_{D,i} \\ \mathsf{T}_{N,i} \end{pmatrix} U_i = \begin{pmatrix} \mathbb{I} & 0 \\ 0 & -\mathbb{I} \end{pmatrix} \begin{pmatrix} \mathsf{T}_{D,j} \\ \mathsf{T}_{N,j} \end{pmatrix} U_j \quad \text{on } \Gamma_{ij} \, , \tag{4}$$

for which we embrace the compact notation $\mathbb{T}_i U_i = \mathbb{X}\,\mathbb{T}_j U_j$ with obvious meanings of the operators $\mathbb{T}_i$ and $\mathbb{X}$.

Remark 1. In fact, multi-trace boundary integral equations were first developed for acoustic and electromagnetic scattering problems and we emphasize that the ideas of this article will naturally apply to them, see [3].

3 Basic Multi-Trace Formulation

For the sake of lucidity, in this section we largely restrict ourselves to the situation $N = 2$, as sketched in Figure 1 for $d = 2$. For the purpose of presenting the local multi-trace formulation this case is generic and completely captures the ideas and essence of the methods.

3.1 Preliminaries

The starting point for deriving multi-trace boundary integral equations is the characterization of traces of local solutions of (1) as the range of a (compound) boundary integral operator known as *Calderón projector*, see [3, Sect. 2.3], [16, Sect. 3.6], and [9, Sect. 5.6]. For the Calderón projector associated with the PDE $\mathsf{L}_i U_i = 0$ on Ω_i we write

$$\mathbb{P}_i : H^{\frac{1}{2}}(\partial\Omega_i) \times H^{-\frac{1}{2}}(\partial\Omega_i) \to H^{\frac{1}{2}}(\partial\Omega_i) \times H^{-\frac{1}{2}}(\partial\Omega_i) \, , \tag{5}$$

and recall that $\mathbb{P}_i$ is connected to the four key boundary integral operators for 2nd-order scalar PDEs according to

$$\mathbb{P}_i = \mathbb{A}_i + \tfrac{1}{2}\mathbb{I} \quad , \quad \mathbb{A}_i = \begin{pmatrix} -\mathsf{K}_i & \mathsf{V}_i \\ \mathsf{W}_i & \mathsf{K}_i' \end{pmatrix} \, , \tag{6}$$

where we have adopted the notations K_i, V_i, W_i, K_i' from [16, Sect. 3.1] for the double layer, single layer, hypersingular, and adjoint double layer boundary integral

[1] As usual, $H(\Delta,\Omega) := \{U \in H^1(\Omega) : \Delta U \in L^2(\Omega)\}$.

operators on $\partial\Omega_i$, respectively. The Calderón projectors owe their importance to the following fundamental theorem [3, Thm. 2.6].

Theorem 1. *If and only if U_i solves $\mathsf{L}_i U_i = 0$ in Ω_i (and satisfies exponential decay conditions at ∞ for $i = 0$), then $(\mathbb{I} - \mathbb{P}_i)\,\mathbb{T}_i U_i = 0$.*

Here, in the interest of compact notation, we relied on the total trace operator $\mathbb{T}_i := \begin{pmatrix} \mathsf{T}_{D,i} \\ \mathsf{T}_{N,i} \end{pmatrix}$. Thus, if U is a solution of (1), we conclude from Theorem 1

$$\begin{aligned} \left(-\mathbb{A}_i + \tfrac{1}{2}\mathbb{I}\right)\mathbb{T}_i U = 0\,, \quad i = 1,2\,, \\ \left(-\mathbb{A}_0 + \tfrac{1}{2}\mathbb{I}\right)\mathbb{T}_0 (U - U_{\mathrm{inc}}) = 0\,. \end{aligned} \tag{7}$$

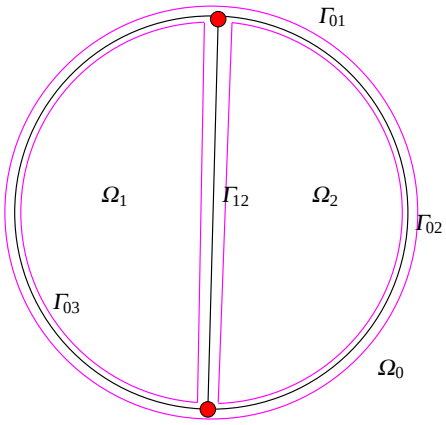


Fig. 1 Geometric situation "$N = 2$" in 2D for derivation of multi-trace boundary integral formulations. Black lines indicate the sub-domain boundaries, magenta lines stand for Cauchy traces, of which there are two on each interface in the multi-trace setting. Red dots mark junction points.

3.2 Derivation

The derivation of the basic MTF casts both (7) and the transmission conditions (4) into weak form. To do so, we need bilinear pairings [2]

$$[\mathfrak{u}_i, \mathfrak{v}_i]_{\partial\Omega_i} := \langle u, \nu\rangle_{\partial\Omega_i} + \langle \nu, \mu\rangle_{\partial\Omega_i}\,, \quad \mathfrak{u}_i := \begin{pmatrix} u \\ \mu \end{pmatrix},\ \mathfrak{v}_i := \begin{pmatrix} \nu \\ \nu \end{pmatrix} \in \mathscr{T}(\partial\Omega_i)\,, \tag{8}$$

on the *local Cauchy trace spaces*[3]

$$\mathscr{T}(\partial\Omega_i) := H^{\frac{1}{2}}(\partial\Omega_i) \times H^{-\frac{1}{2}}(\partial\Omega_i)\,. \tag{9}$$

[2] Fraktur font is used to designate functions in the Cauchy trace space, whereas Roman typeface is reserved for Dirichlet traces, and Greek symbols for Neumann traces.

[3] By Cauchy trace spaces we mean combined Dirichlet and Neumann traces.

In (8), angle brackets designated the bi-linear duality product between $H^{\frac{1}{2}}(\partial\Omega_i)$ and $H^{-\frac{1}{2}}(\partial\Omega_i)$, which reduces to an L^2-pairing for sufficiently regular functions. Then (7) is equivalent to

$$\left[\left(-\mathbb{A}_i+\tfrac{1}{2}\mathbb{I}\right)\mathbb{T}_i U,\mathfrak{v}_i\right]_{\partial\Omega_i}=\text{r.h.s.}\quad\forall\mathfrak{v}_i\in\mathscr{T}(\partial\Omega_i)\;,\;i=0,1,2\;, \tag{10}$$

with "r.h.s.", here and below, representing a linear form on the trial space that provides the excitation.

A possible weak form the transmission conditions (4) can sloppily be stated as

$$\left[\mathbb{T}_i U-\mathbb{X}\,\mathbb{T}_j U,\mathfrak{v}_i|_{\Gamma_{ij}}\right]_{\Gamma_{ij}}=0\quad\forall\text{ “}\mathfrak{v}_i\in\mathscr{T}(\partial\Omega_i)\text{”}\;. \tag{11}$$

The attribute "sloppy" and the quotation marks hint at fundamental problems haunting (11) and those lurk in the failure of the bi-linear pairing $[\cdot,\cdot]_{\Gamma_{ij}}$ to be well defined for restrictions of generic traces to Γ_{ij}.

Temporarily sweeping these difficulties under the rug (and restricting ourselves to the situation $N=2$ illustrated in Figure 1), we now combine (10) and (11) into

$$\begin{aligned}
&\left[\left(\mathbb{A}_0-\tfrac{1}{2}\mathbb{I}\right)\mathbb{T}_0 U,\mathfrak{v}_0\right]_{\partial\Omega_0}-\sigma_{01}\left[\mathbb{T}_0 U-\mathbb{X}\,\mathbb{T}_1 U,\mathfrak{v}_0|_{\Gamma_{01}}\right]_{\Gamma_{01}}\\
&\qquad-\sigma_{02}\left[\mathbb{T}_0 U-\mathbb{X}\,\mathbb{T}_2 U,\mathfrak{v}_0|_{\Gamma_{02}}\right]_{\Gamma_{02}}=\text{r.h.s.}\quad\forall\text{ “}\mathfrak{v}_0\in\mathscr{T}(\partial\Omega_0)\text{”}\;,\\
&\left[\left(\mathbb{A}_1-\tfrac{1}{2}\mathbb{I}\right)\mathbb{T}_1 U,\mathfrak{v}_1\right]_{\partial\Omega_1}-\sigma_{10}\left[\mathbb{T}_1 U-\mathbb{X}\,\mathbb{T}_0 U,\mathfrak{v}_1|_{\Gamma_{10}}\right]_{\Gamma_{10}}\\
&\qquad-\sigma_{12}\left[\mathbb{T}_1 U-\mathbb{X}\,\mathbb{T}_2 U,\mathfrak{v}_1|_{\Gamma_{12}}\right]_{\Gamma_{12}}=\text{r.h.s.}\quad\forall\text{ “}\mathfrak{v}_1\in\mathscr{T}(\partial\Omega_1)\text{”}\;,\\
&\left[\left(\mathbb{A}_2-\tfrac{1}{2}\mathbb{I}\right)\mathbb{T}_2 U,\mathfrak{v}_2\right]_{\partial\Omega_2}-\sigma_{21}\left[\mathbb{T}_2 U-\mathbb{X}\,\mathbb{T}_1 U,\mathfrak{v}_2|_{\Gamma_{21}}\right]_{\Gamma_{21}}\\
&\qquad-\sigma_{20}\left[\mathbb{T}_2 U-\mathbb{X}\,\mathbb{T}_0 U,\mathfrak{v}_2|_{\Gamma_{20}}\right]_{\Gamma_{20}}=\text{r.h.s.}\quad\forall\text{ “}\mathfrak{v}_2\in\mathscr{T}(\partial\Omega_2)\text{”}\;,
\end{aligned} \tag{12}$$

where the σ_{ij} are non-zero weights. These are equations satisfied by the local Cauchy traces $\mathbb{T}_i U$, $i=0,1,2$. Next, we treat these traces as unknowns and call them $\mathfrak{u}_1$, $\mathfrak{u}_2$, and $\mathfrak{u}_3$ which converts (12) into a system of (variational) boundary integral equations. It deserves the label "multi-trace", because the unknowns are separate Cauchy traces for each sub-domain, which yields two pairs of unknown traces on each interface, twice the number used in most other boundary integral formulations, see Figure 1. Adopting a compact notation, (for $N=2$) the problem is posed on the *multi-trace space*

$$\mathscr{M}\mathscr{T}(\Sigma):=\mathscr{T}(\partial\Omega_0)\times\mathscr{T}(\partial\Omega_1)\times\mathscr{T}(\partial\Omega_2)\;. \tag{13}$$

The special variant of (12) proposed in [8] is recovered by setting $\sigma_{ij}=-\frac{1}{2}$. To see, why this is a special choice, note that, for instance,

$$\Big[\mathfrak{u}_0,\,\mathfrak{v}_0|_{\Gamma_{01}}\Big]_{\Gamma_{01}}+\Big[\mathfrak{u}_0,\,\mathfrak{v}_0|_{\Gamma_{02}}\Big]_{\Gamma_{02}}=[\mathfrak{u}_0,\mathfrak{v}_0]_{\partial\Omega_0}\;,\quad \mathfrak{u},\mathfrak{v}\in\mathscr{T}(\partial\Omega_0)\;.$$

Thus, we achieve a massive cancellation of terms and arrive at the *basic multi-trace formulation*: seek $(\mathfrak{u}_0,\mathfrak{u}_1,\mathfrak{u}_2)\in\mathscr{M}\mathscr{T}(\Sigma)$ such that

$$\begin{aligned}
&[\mathbb{A}_0\,\mathfrak{u}_0,\mathfrak{v}_0]_{\partial\Omega_0}-\tfrac{1}{2}\Big[\,\mathbb{X}\,\mathfrak{u}_1|_{\Gamma_{01}},\mathfrak{v}_0|_{\Gamma_{01}}\Big]_{\Gamma_{01}}-\tfrac{1}{2}\Big[\,\mathbb{X}\,\mathfrak{u}_2|_{\Gamma_{02}},\mathfrak{v}_0|_{\Gamma_{02}}\Big]_{\Gamma_{02}}=\text{r.h.s.}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\forall\text{ “}\mathfrak{v}_0\in\mathscr{T}(\partial\Omega_0)\text{”}\,,\\
&[\mathbb{A}_1\,\mathfrak{u}_1,\mathfrak{v}_1]_{\partial\Omega_1}-\tfrac{1}{2}\Big[\,\mathbb{X}\,\mathfrak{u}_0|_{\Gamma_{10}},\mathfrak{v}_1|_{\Gamma_{10}}\Big]_{\Gamma_{10}}-\tfrac{1}{2}\Big[\,\mathbb{X}\,\mathfrak{u}_2|_{\Gamma_{12}},\mathfrak{v}_1|_{\Gamma_{12}}\Big]_{\Gamma_{12}}=\text{r.h.s.}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\forall\text{ “}\mathfrak{v}_1\in\mathscr{T}(\partial\Omega_1)\text{”}\,,\\
&[\mathbb{A}_2\,\mathfrak{u}_2,\mathfrak{v}_2]_{\partial\Omega_2}-\tfrac{1}{2}\Big[\,\mathbb{X}\,\mathfrak{u}_1|_{\Gamma_{21}},\mathfrak{v}_2|_{\Gamma_{21}}\Big]_{\Gamma_{21}}-\tfrac{1}{2}\Big[\,\mathbb{X}\,\mathfrak{u}_0|_{\Gamma_{20}},\mathfrak{v}_2|_{\Gamma_{20}}\Big]_{\Gamma_{20}}=\text{r.h.s.}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\forall\text{ “}\mathfrak{v}_2\in\mathscr{T}(\partial\Omega_2)\text{”}\,,
\end{aligned}\tag{14}$$

where, again, the quotation marks acknowledge difficulties besetting the use of generic traces as trial and test functions. The variational formulations for general N can be found in [3, Sect. 6] and [8, Sect. 3.2.3].

3.3 Analysis

Let us take a closer look at the coupling terms in (14). For $\mathfrak{u}_i\in\mathscr{T}(\partial\Omega_i)$ and $\mathfrak{v}_j\in\mathscr{T}(\partial\Omega_j)$ we find

$$\mathbb{X}\,\mathfrak{u}_i|_{\Gamma_{ij}}\,,\mathfrak{v}_j\big|_{\Gamma_{ij}}\in H^{\frac{1}{2}}(\Gamma_{ij})\times H^{-\frac{1}{2}}(\Gamma_{ij})\;.$$

Unfortunately, $H^{\frac{1}{2}}(\Gamma_{ij})$ and $H^{-\frac{1}{2}}(\Gamma_{ij})$ are not in duality with pivot space $L^2(\Gamma_{ij})$. More precisely, $(\mathfrak{u}_i,\mathfrak{v}_j)\mapsto\Big[\,\mathbb{X}\,\mathfrak{u}_i|_{\Gamma_{ij}},\,\mathfrak{v}_j\big|_{\Gamma_{ij}}\Big]_{\Gamma_{ij}}$ is not bounded on $\mathscr{T}(\partial\Omega_i)\times\mathscr{T}(\partial\Omega_j)$, which renders (14) meaningless without the quotation marks.

As a remedy, more regular test functions have to be used, namely functions whose restrictions to Γ_{ij} belong to the $L^2(\Gamma_{ij})$-dual of $H^{\frac{1}{2}}(\Gamma_{ij})\times H^{-\frac{1}{2}}(\Gamma_{ij})$, which is known to coincide with $\widetilde{H}^{\frac{1}{2}}(\Gamma_{ij})\times\widetilde{H}^{-\frac{1}{2}}(\Gamma_{ij})$, where the latter spaces are spaces of functions, whose extensions by zero from Γ_{ij} to $\partial\Omega_j$ are still valid functions in $H^{\frac{1}{2}}(\partial\Omega_j)\times H^{-\frac{1}{2}}(\partial\Omega_j)$. We remind that $\widetilde{H}^{\frac{1}{2}}(\Gamma_{ij})\times\widetilde{H}^{-\frac{1}{2}}(\Gamma_{ij})$ is a *dense* subspace of $H^{\frac{1}{2}}(\Gamma_{ij})\times H^{-\frac{1}{2}}(\Gamma_{ij})$ with *strictly stronger norm*, see [11, Ch. 3] and [8, Sect. 2]. Thus, proper test spaces in (14) are

$$\widetilde{\mathscr{T}}(\partial\Omega_j)=\bigotimes_{i\neq j}\widetilde{H}^{\frac{1}{2}}(\Gamma_{ij})\times\widetilde{H}^{-\frac{1}{2}}(\Gamma_{ij})\;,\quad j=0,1,2\;,\tag{15}$$

since the bilinear form m associated with (14) turns out to be bounded as a mapping

$$\mathsf{m} : \mathscr{M}\mathscr{T}(\Sigma) \times \widetilde{\mathscr{M}\mathscr{T}}(\Sigma) \to \mathbb{R} \,,$$

where $\widetilde{\mathscr{M}\mathscr{T}}(\Sigma)$ is defined in analogy to (13) this time based on $\widetilde{\mathscr{T}}(\partial\Omega_j)$.

A key observation concerns the *block skew-symmetric* structure of (14) due to

$$\Big[\, \mathbb{X}\,\mathfrak{u}_i|_{\Gamma_{ij}}, \mathfrak{v}_j\big|_{\Gamma_{ij}} \Big]_{\Gamma_{ij}} = -\Big[\, \mathbb{X}\,\mathfrak{v}_j\big|_{\Gamma_{ij}}, \mathfrak{u}_i|_{\Gamma_{ij}} \Big]_{\Gamma_{ij}} \,, \quad \begin{array}{l} \mathfrak{u}_i \in \widetilde{\mathscr{T}}(\partial\Omega_i), \\ \mathfrak{v}_j \in \widetilde{\mathscr{T}}(\partial\Omega_j) \,. \end{array} \tag{16}$$

In light of the well known ellipticity of the boundary integral operators [16, Sect. 3.5.1]

$$\exists C > 0 : \quad |\,[\mathbb{A}_j\,\mathfrak{v}_j, \mathfrak{v}_j]_{\partial\Omega_j}\,| \geq C \left\|\mathfrak{v}_j\right\|^2_{\mathscr{T}(\partial\Omega_j)} \quad \forall \mathfrak{v}_j \in \mathscr{T}(\partial\Omega_j) \,, \tag{17}$$

(16) immediately implies the $\mathscr{M}\mathscr{T}(\Sigma)$-ellipticity of m:

$$\exists C > 0 : \quad |\,\mathsf{m}(\overrightarrow{\mathfrak{v}}, \overrightarrow{\mathfrak{v}})| \geq C \left\|\overrightarrow{\mathfrak{v}}\right\|^2_{\mathscr{M}\mathscr{T}(\Sigma)} \quad \forall \overrightarrow{\mathfrak{v}} \in \widetilde{\mathscr{M}\mathscr{T}}(\Sigma) \,. \tag{18}$$

From (18) we conclude existence and uniqueness of solutions of (14) with trial space $\widetilde{\mathscr{M}\mathscr{T}}(\Sigma)$. Not straightforwardly, however, because the lack of continuity of m on $\mathscr{M}\mathscr{T}(\Sigma) \times \mathscr{M}\mathscr{T}(\Sigma)$ bars us from appealing to the Riesz representation theorem. Fortunately, as elaborated in [8, Sect. 3.2.8], we can rely a result by J.L. Lions [10, Ch. III, Thm. 1.1] along with the density of $\widetilde{\mathscr{M}\mathscr{T}}(\Sigma)$ in $\mathscr{M}\mathscr{T}(\Sigma)$:

Theorem 2. *The variational problem* (14) *on* $\mathscr{M}\mathscr{T}(\Sigma) \times \widetilde{\mathscr{M}\mathscr{T}}(\Sigma)$ *possesses a unique solution in* $\mathscr{M}\mathscr{T}(\Sigma)$ *that depends continuously on the right hand side.*

Remark 2. The result of Theorem 2 crucially hinges on the ellipticity (18), which can be taken for granted only for the choice $\sigma_{ij} = -\frac{1}{2}$. For general weights σ_{ij} existence and uniqueness of solutions of (12) is an open problem.

Remark 3. For scattering problems the sesqui-linear form of (14) will be merely coercive. In this case uniqueness of solutions has to be established by other arguments, see [8, Sect. 3.2.6], and existence follows from Fredholm theory.

4 Transformed Multi-Trace Formulations

4.1 Optimal transmission conditions

An important motivation for the development of multi-trace BIE was the desire to obtain linear systems of equations that readily lend themselves to additive Schwarz ("block Jacobi") preconditioning. On the level of the transmission problem (1), this amounts to solving local boundary value problems on Ω_i using Dirichlet or Neumann boundary data from the previous iterates on the adjacent sub-domains. However, the transmission conditions (1b) may not lead to satisfactory convergence.

To understand how alternative transmission conditions can boost an additive Schwarz iteration, let us examine the very simple situation with $N = 1$, $\Sigma = \Gamma := \partial\Omega_0 = \partial\Omega_1$. There is a special transmission condition that effects convergence in one step! To state it, we introduce the Dirichlet-to-Neumann (DtN) operators

$$\mathsf{DtN}_0, \mathsf{DtN}_1 : H^{\frac{1}{2}}(\Gamma) \to H^{-\frac{1}{2}}(\Gamma) \tag{19}$$

and their inverses, the Neumann-to-Dirichlet (NtD) operators

$$\mathsf{NtD}_0, \mathsf{NtD}_1 : H^{-\frac{1}{2}}(\Gamma) \to H^{\frac{1}{2}}(\Gamma) \quad , \quad \mathsf{NtD}_i = \mathsf{DtN}_i^{-1} . \tag{20}$$

The subscript indicates whether they are associated with a boundary value problem $\mathsf{L}_i U = 0$ on Ω_0 or Ω_1, respectively. Recall that DtN operators, sometimes called Steklov-Poincaré operators, return the Neumann trace of a solution of a boundary value problem for prescribed Dirichlet data [11, Ch. 4]. The DtN operators associated with bounded subdomains are linear, but DtN_0 is merely affine due to the "nonzero boundary condition at infinity" imposed through U_{inc}. In any case, the linear parts of the operators DtN_i and NtD_i are symmetric and positive.

Based on these operators, we introduce modified transmission conditions across Γ:

$$\mathsf{T}_{D,1} U - \mathsf{NtD}_1(\mathsf{T}_{N,1} U) = \mathsf{T}_{D,0} U + \mathsf{NtD}_1(\mathsf{T}_{N,0} U) , \tag{21a}$$
$$\mathsf{DtN}_0(\mathsf{T}_{D,1} U) + \mathsf{T}_{N,1} U = \mathsf{DtN}_0(\mathsf{T}_{D,0} U) - \mathsf{T}_{N,0} U . \tag{21b}$$

These transmission conditions are perfectly symmetric with respect to Ω_0 and Ω_1, since, thanks to $\mathsf{NtD}_i = \mathsf{DtN}_i^{-1}$, we can rewrite (21) in the equivalent form

$$\mathsf{DtN}_1(\mathsf{T}_{D,1} U) - \mathsf{T}_{N,1} U = \mathsf{DtN}_1(\mathsf{T}_{D,0} U) + \mathsf{T}_{N,0} U , \tag{22a}$$
$$\mathsf{T}_{D,1} U + \mathsf{NtD}_0(\mathsf{T}_{N,1} U) = \mathsf{T}_{D,0} U - \mathsf{NtD}_0(\mathsf{T}_{N,0} U) . \tag{22b}$$

Invertibility of the involved operators yields another equivalence

$$(21) \quad \Leftrightarrow \quad (22) \quad \Leftrightarrow \quad \begin{cases} \mathsf{T}_{D,1} U = \mathsf{T}_{D,0} U , \\ \mathsf{T}_{N,1} U = -\mathsf{T}_{N,0} U , \end{cases} \tag{23}$$

which confirms that the original transmission conditions (4) are implied by our modified versions.

Following the policy of Section 3.2, we aim for an MTF based on (21) and first cast the transmission conditions into weak form

$$[(\mathbb{I} + \mathbb{M})\,\mathbb{T}_1 U - (\mathbb{I} + \mathbb{M})\,\mathbb{X}(\mathbb{T}_0 U), \mathfrak{v}]_\Gamma = 0 \quad \forall \mathfrak{v} \in \mathscr{T}(\Gamma) , \tag{24}$$
$$\Updownarrow$$
$$[(\mathbb{I} - \mathbb{M})\,\mathbb{T}_0 U - (\mathbb{I} - \mathbb{M})\,\mathbb{X}(\mathbb{T}_1 U), \mathfrak{v}]_\Gamma = 0 \quad \forall \mathfrak{v} \in \mathscr{T}(\Gamma) , \tag{25}$$

with an affine linear operator

$$\mathbb{M} := \begin{pmatrix} 0 & -\mathsf{NtD}_1 \\ \mathsf{DtN}_0 & 0 \end{pmatrix} : \mathscr{T}(\Gamma) \to \mathscr{T}(\Gamma) \,. \tag{26}$$

Note that in the above manipulations, we have used $\mathbb{X}\,\mathbb{M} = -\mathbb{M}\,\mathbb{X}$. This yields the generalized multi-trace formulation: seek $\mathfrak{u}_0, \mathfrak{u}_1 \in \mathscr{T}(\Gamma)$ such that

$$\left[\left(-\mathbb{A}_0 + \tfrac{1}{2}\mathbb{I}\right)\mathfrak{u}_0, \mathfrak{v}\right]_\Gamma + \sigma_{01}\left[(\mathbb{I}-\mathbb{M})\mathfrak{u}_0 - (\mathbb{I}-\mathbb{M})\,\mathbb{X}\,\mathfrak{u}_1, \mathfrak{v}\right]_\Gamma = 0 \,, \tag{27a}$$

$$\sigma_{10}\left[(\mathbb{I}+\mathbb{M})\mathfrak{u}_1 - (\mathbb{I}+\mathbb{M})\,\mathbb{X}\,\mathfrak{u}_0, \mathfrak{v}\right]_\Gamma + \left[\left(-\mathbb{A}_1 + \tfrac{1}{2}\mathbb{I}\right)\mathfrak{u}_1, \mathfrak{v}\right]_\Gamma = 0 \,, \tag{27b}$$

for all $\mathfrak{v} \in \mathscr{T}(\Gamma)$. Again, we may go after cancellation by setting $\sigma_{01} = \sigma_{10} = -\frac{1}{2}$, so that (27a) is simplified to: seek $\mathfrak{u}_0, \mathfrak{u}_1 \in \mathscr{T}(\Gamma)$ such that

$$-\left[(\mathbb{A}_0 - \tfrac{1}{2}\mathbb{M})\mathfrak{u}_0, \mathfrak{v}\right]_\Gamma + \tfrac{1}{2}\left[(\mathbb{I}-\mathbb{M})\,\mathbb{X}\,\mathfrak{u}_1, \mathfrak{v}\right]_\Gamma = 0 \,, \tag{28a}$$

$$\tfrac{1}{2}\left[(\mathbb{I}+\mathbb{M})\,\mathbb{X}\,\mathfrak{u}_0, \mathfrak{v}\right]_\Gamma - \left[(\mathbb{A}_1 + \tfrac{1}{2}\mathbb{M})\mathfrak{u}_1, \mathfrak{v}\right]_\Gamma = 0 \,, \tag{28b}$$

for all $\mathfrak{v} \in \mathscr{T}(\Gamma)$. This linear variational problem may be solved by means of the following (undamped) additive Schwarz method: given approximations $\mathfrak{u}_0^{(k)}, \mathfrak{u}_1^{(k)} \in \mathscr{T}(\Gamma)$, $k = 0, 1, \ldots$, compute $\mathfrak{u}_0^{(k+1)}, \mathfrak{u}_1^{(k+1)} \in \mathscr{T}(\Gamma)$ as solutions of

$$-\left[(\mathbb{A}_0 - \tfrac{1}{2}\mathbb{M})\mathfrak{u}_0^{(k+1)}, \mathfrak{v}\right]_\Gamma + \tfrac{1}{2}\left[(\mathbb{I}-\mathbb{M})\,\mathbb{X}\,\mathfrak{u}_1^{(k)}, \mathfrak{v}\right]_\Gamma = 0 \,, \tag{29a}$$

$$\tfrac{1}{2}\left[(\mathbb{I}+\mathbb{M})\,\mathbb{X}\,\mathfrak{u}_0^{(k)}, \mathfrak{v}\right]_\Gamma - \left[(\mathbb{A}_1 + \tfrac{1}{2}\mathbb{M})\mathfrak{u}_1^{(k+1)}, \mathfrak{v}\right]_\Gamma = 0 \,. \tag{29b}$$

$\forall \mathfrak{v} \in \mathscr{T}(\Gamma)$

Lemma 1. *Assuming unique solvability of the linear variational problem* (29)*, and* $\mathfrak{u}_0^{(0)} = \mathfrak{u}_1^{(0)} = 0$*, the iteration will become stationary after one step, with* $\mathbb{T}_0 U = \mathfrak{u}_0^{(1)}$ *and* $\mathbb{T}_1 U = \mathfrak{u}_1^{(1)}$*, where* U *is the solution of the transmission problem* (1)*.*

Proof. Consider the boundary value problem posed on Ω_0:

$$-\operatorname{div}(\mu_0 \,\mathbf{grad}\, U^{(k+1)}) + U^{(k+1)} = 0 \quad \text{in } \Omega_0 \,, \tag{30a}$$

$$\mathsf{DtN}_1(\mathsf{T}_{D,0} U^{(k+1)}) + \mathsf{T}_{N,0} U^{(k+1)} = \mathsf{DtN}_1(\mathsf{T}_{D,1} U^{(k)}) - \mathsf{T}_{N,1} U^{(k)} \quad \text{on } \Gamma \,, \tag{30b}$$

$$\mathsf{DtN}_0(\mathsf{T}_{D,0} U^{(k+1)}) - \mathsf{T}_{N,0} U^{(k+1)} = \mathsf{DtN}_0(\mathsf{T}_{D,1} U^{(k)}) + \mathsf{T}_{N,1} U^{(k)} \quad \text{on } \Gamma \,, \tag{30c}$$

$$U^{(k+1)} - U_{\text{inc}} \quad \text{satisfies decay conditions at } \infty \,, \tag{30d}$$

and assume that it has a solution. Then, recalling Theorem 1 and the definition of $\mathbb{M}$, we find that with $\mathfrak{u}_1^{(k)} := \mathbb{T}_1 U^{(k)}$ the Cauchy traces $\mathfrak{u}_0^{(k+1)} := \mathbb{T}_0 U^{(k+1)}$ provide a solution of (29a). However, in general (30) will fail to be a meaningful boundary value problem, because too many boundary conditions are imposed on Γ. Yet, if $U^{(k)} = 0$, then the boundary conditions (30b) and (30c) become

$$\mathsf{DtN}_1(\mathsf{T}_{D,0} U^{(1)}) + \mathsf{T}_{N,0} U^{(1)} = 0 \quad \text{on } \Gamma \,, \tag{31a}$$

$$\mathsf{DtN}_0(\mathsf{T}_{D,0} U^{(1)}) - \mathsf{T}_{N,0} U^{(1)} = \mathsf{DtN}_0(0) \quad \text{on } \Gamma \,. \tag{31b}$$

Notice that (31b) is redundant, satisfied by *any* solution of (30a) complying with (30d). What remains in terms of effective boundary conditions on Γ is (31a), which represents a well-posed impedance boundary condition und guarantees the existence of a unique solution $U^{(k+1)}$. The Cauchy trace $\mathfrak{u}_0^{(1)} := \mathbb{T}_0 U^{(k)}$ of that solution will satisfy

$$-\left[(\mathbb{A}_0 - \tfrac{1}{2}\mathbb{M})\mathfrak{u}_0^{(1)}, \mathfrak{v}\right]_\Gamma = \tfrac{1}{2}\left[\begin{pmatrix} 0 \\ \mathsf{DtN}_0(0) \end{pmatrix}, \mathfrak{v}\right]_\Gamma , \tag{32}$$

which agrees with the variational problem (29a) to be solved in the first step of the Schwarz iteration with initial guess $\mathfrak{u}_1^{(0)} = 0$.

Similar considerations apply to (29b). Here we start from the boundary value problem with redundant boundary conditions

$$-\operatorname{div}(\mu_1 \mathbf{grad} U^{(k+1)}) + U^{(k+1)} = 0 \quad \text{in } \Omega_1 , \tag{33a}$$

$$\mathsf{DtN}_0(\mathsf{T}_{D,1} U^{(k+1)}) + \mathsf{T}_{N,1} U^{(k+1)} = \mathsf{DtN}_0(\mathsf{T}_{D,0} U^{(k)}) - \mathsf{T}_{N,0} U^{(k)} 0 \quad \text{on } \Gamma , \tag{33b}$$

$$\mathsf{DtN}_1(\mathsf{T}_{D,1} U^{(k+1)}) - \mathsf{T}_{N,1} U^{(k+1)} = \mathsf{DtN}_1(\mathsf{T}_{D,0} U^{(k)}) + \mathsf{T}_{N,0} U^{(k)} \quad \text{on } \Gamma . \tag{33c}$$

If this has a solution $u^{(k+1)}$, its Cauchy trace $\mathfrak{u}_1^{(k+1)} := \mathbb{T}_1 U^{(k+1)}$ will solve (29b) provided that $\mathfrak{u}_0^{(k)} := \mathbb{T}_0 U^{(k)}$. Again, if $U^{(k)} = 0$, the boundary conditions on Γ are converted into

$$\mathsf{DtN}_0(\mathsf{T}_{D,1} U^{(1)}) + \mathsf{T}_{N,1} U^{(1)} = \mathsf{DtN}_0(0) \quad \text{on } \Gamma , \tag{34a}$$

$$\mathsf{DtN}_1(\mathsf{T}_{D,1} U^{(1)}) - \mathsf{T}_{N,1} U^{(1)} = 0 \quad \text{on } \Gamma , \tag{34b}$$

and the second is always fulfilled and can be dropped. This results in a well posed elliptic boundary value problem and the Cauchy trace $\mathfrak{u}_1^{(1)} := \mathbb{T}_1 U^{(k+1)}$ solves

$$\left[(\mathbb{A}_1 + \tfrac{1}{2}\mathbb{M})\mathfrak{u}_1^{(1)}, \mathfrak{v}\right]_\Gamma = \tfrac{1}{2}\left[\begin{pmatrix} 0 \\ \mathsf{DtN}_0(0) \end{pmatrix}, \mathfrak{v}\right]_\Gamma , \tag{35}$$

which amounts to the second linear problem faced in the first step of the Schwarz method (29) starting from zero.

By the definition of the Dirichlet-to-Neumann operators, the combined solutions of the boundary value problems (30a), (31a), (30d) and (33a), (34a) provide a solution of the transmission problem (1). Thus $\mathfrak{u}_0^{(1)}$ and $\mathfrak{u}_1^{(1)}$ from (32) and (35) are the Cauchy traces of that solution. Here we rely on the assumption of the Lemma that ensures uniqueness of $\mathfrak{u}_0^{(1)}$ and $\mathfrak{u}_1^{(1)}$. Thus they are the desired final solutions and the Schwarz iteration will become stationary after one step. □

As a consequence of this Lemma, the additive Schwarz iteration (29) converges after two steps, thanks to the transmission conditions (21)/(22), which we call "optimal" for this reason. Unfortunately, the "optimal transmission conditions" destroy positivity of the resulting multi-trace operator, which turned out a key property in

Section 3.3, see (18). We still find

$$[(\mathbb{I}-\mathbb{M})\,\mathbb{X}\,\mathfrak{v}_1,\mathfrak{v}_0]_\Gamma = -\,[(\mathbb{I}+\mathbb{M})\,\mathbb{X}\,\mathfrak{v}_0,\mathfrak{v}_1]_\Gamma \quad \forall \mathfrak{v}_0,\mathfrak{v}_1 \in \mathscr{T}(\partial\Omega)\,,$$

but the ellipticity of the diagonal operators, e.g.,

$$\mathbb{A}_0 - \tfrac{1}{2}\mathbb{M} = \begin{pmatrix} -\mathsf{K}_0 & \mathsf{V}_0 + \frac{1}{2}\mathsf{NtD}_1 \\ \mathsf{W}_0 - \frac{1}{2}\mathsf{DtN}_0 & \mathsf{K}_0' \end{pmatrix}\,, \tag{36}$$

is lost. Hence, rigorous results about existence and uniqueness of solutions of (28) are still missing even in the case $N = 1$. This is an open problem for future research.

Moreover, the optimal transmission conditions (21) require the realization of DtN and NtD operators. Their exact implementation is not an option for practical schemes. Thus, in the next section we consider local approximations for the optimal transmission conditions.

4.2 Local impedance transmission conditions

The considerations of the previous section suggest that for $N > 1$ we use transmission conditions similar to (21) *locally* on the interface Γ_{ij}, where $\mathsf{DtN}_j, \mathsf{DtN}_i$ etc. are replaced by suitable approximations. The resulting so-called local impedance transmission conditions across the interface Γ_{ij} can be written in the form

$$\mathsf{B}_{ij}(\mathsf{T}_{D,i}U) + \mathsf{T}_{N,i}U = \mathsf{B}_{ij}(\mathsf{T}_{D,j}U) - \mathsf{T}_{N,j}U\,, \tag{37a}$$
$$\mathsf{B}_{ji}(\mathsf{T}_{D,i}U) - \mathsf{T}_{N,i}U = \mathsf{B}_{ji}(\mathsf{T}_{D,j}U) + \mathsf{T}_{N,j}U\,. \tag{37b}$$

where B_{ij} and B_{ji} are invertible (affine) linear operators of "DtN-type" mapping $H^{\frac{1}{2}}(\Gamma_{ij})$ onto $H^{-\frac{1}{2}}(\Gamma_{ij})$. Parallel to the switch from (21) to (22), invertibility of the involved operators yields another equivalence

$$\mathsf{T}_{D,i}U + \mathsf{C}_{ij}(\mathsf{T}_{N,i}U) = \mathsf{T}_{D,j}U - \mathsf{C}_{ij}(\mathsf{T}_{N,j}U)\,, \tag{38a}$$
$$\mathsf{T}_{D,i}U - \mathsf{C}_{ji}(\mathsf{T}_{N,i}U) = \mathsf{T}_{D,j}U + \mathsf{C}_{ji}(\mathsf{T}_{N,j}U)\,. \tag{38b}$$

where $\mathsf{C}_{ij} = \mathsf{B}_{ij}^{-1} : H^{-\frac{1}{2}}(\Gamma_{ij}) \to H^{\frac{1}{2}}(\Gamma_{ij})$ and $\mathsf{C}_{ji} = \mathsf{B}_{ji}^{-1} : H^{-\frac{1}{2}}(\Gamma_{ij}) \to H^{\frac{1}{2}}(\Gamma_{ij})$. We can then write the weak form of the local impedance transmission conditions as:

$$[(\mathbb{I}+\mathbb{S}_{ij})\,\mathbb{T}_j U - (\mathbb{I}+\mathbb{S}_{ij})\,\mathbb{X}(\mathbb{T}_i U),\mathfrak{v}]_{\Gamma_{ij}} = 0 \quad \forall \mathfrak{v} \in \widetilde{\mathscr{T}}(\Gamma_{ij})\,, \tag{39}$$

$$\Updownarrow$$

$$[(\mathbb{I}-\mathbb{S}_{ij})\,\mathbb{T}_i U - (\mathbb{I}-\mathbb{S}_{ij})\,\mathbb{X}(\mathbb{T}_j U),\mathfrak{v}]_{\Gamma_{ij}} = 0 \quad \forall \mathfrak{v} \in \widetilde{\mathscr{T}}(\Gamma_{ij})\,, \tag{40}$$

with an affine linear operator

$$\mathbb{S}_{ij} := \begin{pmatrix} 0 & \mathsf{C}_{ij} \\ -\mathsf{B}_{ji} & 0 \end{pmatrix} : \mathscr{T}(\Gamma_{ij}) \to \mathscr{T}(\Gamma_{ij}) \,. \tag{41}$$

Retracing the steps detailed in Section 3.2 based on (39), we end up with the *local multi-trace variational problem*, here stated for $N = 2$: seek $(\mathfrak{u}_0, \mathfrak{u}_1, \mathfrak{u}_2) \in \mathscr{M}\mathscr{T}(\Sigma)$ such that

$$\begin{aligned}
&[\mathbb{A}_0\,\mathfrak{u}_0, \mathfrak{v}_0]_{\partial\Omega_0} + \tfrac{1}{2}\,[\mathbb{S}_{01}\mathfrak{u}_0, \mathfrak{v}_0]_{\Gamma_{01}} + \tfrac{1}{2}\,[\mathbb{S}_{02}\mathfrak{u}_0, \mathfrak{v}_0]_{\Gamma_{02}} - \\
&\qquad \tfrac{1}{2}\,[(\mathbb{I}+\mathbb{S}_{01})\,\mathbb{X}\,\mathfrak{u}_1, \mathfrak{v}_0]_{\Gamma_{01}} - \tfrac{1}{2}\,[(\mathbb{I}+\mathbb{S}_{02})\,\mathbb{X}\,\mathfrak{u}_2, \mathfrak{v}_0]_{\Gamma_{02}} = 0\,, \\
&[\mathbb{A}_1\,\mathfrak{u}_1, \mathfrak{v}_1]_{\partial\Omega_1} + \tfrac{1}{2}\,[\mathbb{S}_{10}\mathfrak{u}_1, \mathfrak{v}_1]_{\Gamma_{01}} + \tfrac{1}{2}\,[\mathbb{S}_{12}\mathfrak{u}_1, \mathfrak{v}_1]_{\Gamma_{12}} - \\
&\qquad \tfrac{1}{2}\,[(\mathbb{I}+\mathbb{S}_{10})\,\mathbb{X}\,\mathfrak{u}_0, \mathfrak{v}_1]_{\Gamma_{01}} - \tfrac{1}{2}\,[(\mathbb{I}+\mathbb{S}_{12})\,\mathbb{X}\,\mathfrak{u}_2, \mathfrak{v}_1]_{\Gamma_{12}} = 0\,, \\
&[\mathbb{A}_2\,\mathfrak{u}_2, \mathfrak{v}_2]_{\partial\Omega_2} + \tfrac{1}{2}\,[\mathbb{S}_{20}\mathfrak{u}_2, \mathfrak{v}_2]_{\Gamma_{02}} + \tfrac{1}{2}\,[\mathbb{S}_{21}\mathfrak{u}_2, \mathfrak{v}_2]_{\Gamma_{12}} - \\
&\qquad \tfrac{1}{2}\,[(\mathbb{I}+\mathbb{S}_{20})\,\mathbb{X}\,\mathfrak{u}_0, \mathfrak{v}_2]_{\Gamma_{02}} - \tfrac{1}{2}\,[(\mathbb{I}+\mathbb{S}_{21})\,\mathbb{X}\,\mathfrak{u}_1, \mathfrak{v}_2]_{\Gamma_{12}} = 0\,,
\end{aligned} \tag{42}$$

for all $(\mathfrak{v}_1, \mathfrak{v}_2, \mathfrak{v}_3) \in \widetilde{\mathscr{M}\mathscr{T}}(\Sigma)$. Of course, local pairings on interfaces involve restrictions onto those interfaces even if not apparent from the notation. As explained in Section 3.3, this entails using the more regular test space $\widetilde{\mathscr{M}\mathscr{T}}(\Sigma)$.

An additive Schwarz method analogous to (29) may be applied to (42) as an iterative solver or preconditioner. The corresponding undamped iteration seeks $(\mathfrak{u}_0^{(k+1)}, \mathfrak{u}_1^{(k+1)}, \mathfrak{u}_2^{(k+1)}) \in \mathscr{M}\mathscr{T}(\Sigma)$ such that

$$\begin{aligned}
&\left[\mathbb{A}_0\,\mathfrak{u}_0^{(k+1)}, \mathfrak{v}_0\right]_{\partial\Omega_0} + \tfrac{1}{2}\left[\mathbb{S}_{01}\mathfrak{u}_0^{(k+1)}, \mathfrak{v}_0\right]_{\Gamma_{01}} + \tfrac{1}{2}\left[\mathbb{S}_{02}\mathfrak{u}_0^{(k+1)}, \mathfrak{v}_0\right]_{\Gamma_{02}} - \\
&\qquad \tfrac{1}{2}\left[(\mathbb{I}+\mathbb{S}_{01})\,\mathbb{X}\,\mathfrak{u}_1^{(k)}, \mathfrak{v}_0\right]_{\Gamma_{01}} - \tfrac{1}{2}\left[(\mathbb{I}+\mathbb{S}_{02})\,\mathbb{X}\,\mathfrak{u}_2^{(k)}, \mathfrak{v}_0\right]_{\Gamma_{02}} = 0\,, \\
&\left[\mathbb{A}_1\,\mathfrak{u}_1^{(k+1)}, \mathfrak{v}_1\right]_{\partial\Omega_1} + \tfrac{1}{2}\left[\mathbb{S}_{10}\mathfrak{u}_1^{(k+1)}, \mathfrak{v}_1\right]_{\Gamma_{01}} + \tfrac{1}{2}\left[\mathbb{S}_{12}\mathfrak{u}_1^{(k+1)}, \mathfrak{v}_1\right]_{\Gamma_{12}} - \\
&\qquad \tfrac{1}{2}\left[(\mathbb{I}+\mathbb{S}_{10})\,\mathbb{X}\,\mathfrak{u}_0^{(k)}, \mathfrak{v}_1\right]_{\Gamma_{01}} - \tfrac{1}{2}\left[(\mathbb{I}+\mathbb{S}_{12})\,\mathbb{X}\,\mathfrak{u}_2^{(k+1)}, \mathfrak{v}_1\right]_{\Gamma_{12}} = 0\,, \\
&\left[\mathbb{A}_2\,\mathfrak{u}_2^{(k+1)}, \mathfrak{v}_2\right]_{\partial\Omega_2} + \tfrac{1}{2}\left[\mathbb{S}_{20}\mathfrak{u}_2^{(k+1)}, \mathfrak{v}_2\right]_{\Gamma_{02}} + \tfrac{1}{2}\left[\mathbb{S}_{21}\mathfrak{u}_2^{(k+1)}, \mathfrak{v}_2\right]_{\Gamma_{12}} - \\
&\qquad \tfrac{1}{2}\left[(\mathbb{I}+\mathbb{S}_{20})\,\mathbb{X}\,\mathfrak{u}_0^{(k)}, \mathfrak{v}_2\right]_{\Gamma_{02}} - \tfrac{1}{2}\left[(\mathbb{I}+\mathbb{S}_{21})\,\mathbb{X}\,\mathfrak{u}_1^{(k)}, \mathfrak{v}_2\right]_{\Gamma_{12}} = 0\,,
\end{aligned} \tag{43}$$

for all $(\mathfrak{v}_1, \mathfrak{v}_2, \mathfrak{v}_3) \in \widetilde{\mathscr{M}\mathscr{T}}(\Sigma)$, where a superscript (k) indicates the use of approximations from the previous iteration. As is clear from the considerations of Section 4.1 the choice of B_i, B_j will directly affect the convergence of the Schwarz iteration applied to the multi-trace variational problem. A systematic study still has to be conducted.

Remark 4. So far, the development and analysis of multi-trace methods have focused on acoustic and electromagnetic *wave propagation problems*, see [3, Sect. 1.2]. There the simplest choice for approximate local Dirichlet-to-Neumann operators

seems to be a first order complex Robin transmission condition (TC), introduced in [4], where the operators are chosen in the form

$$\mathsf{B}_{ij} = \mathsf{B}_{ji} = -\eta_{ij}\imath\kappa\,, \quad \eta_{ij} \in \mathbb{R}\,. \tag{44}$$

This choice makes the Schwarz iteration converge quickly for propagating eigenmodes, though the evanescent modes fail to converge. Further work has sought to improve the Robin TCs to ensure convergence of both propagating and evanescent modes [2, 1]. Of particular interest are the so-called optimized Schwarz methods, where the coefficients used in the transmission conditions are obtained by solving min-max optimization problems for half-space model problems. These include the optimized Schwarz method with two-sided Robin TCs [7] and optimized second order transmission conditions [6]. Schwarz methods with high order transmission conditions have also been developed for high frequency time-harmonic Maxwell's Equations. We mention recent works [5] and [12]. The former one is based on the optimized Schwarz methods. The latter develops a true second order TC together with a global plane wave deflation technique to further improve the convergence for electrically large problems.

References

1. Boubendir, Y., Antoine, X., Geuzaine, C.: A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation. J. Comput. Phys. **231**(2), 262–280 (2012).
2. Boubendir, Y., Bendali, A., Fares, M.B.: Coupling of a non-overlapping domain decomposition method for a nodal finite element method with a boundary element method. Internat. J. Numer. Methods Engrg. **73**(11), 1624–1650 (2008).
3. Claeys, X., Hiptmair, R., Jerez-Hanckes, C.: Multi-trace boundary integral equations. Report 2012-20, SAM, ETH Zürich, Zürich, Switzerland (2012). Contribution to "Direct and Inverse Problems in Wave Propagation and Applications", I. Graham, U. Langer, M. Sini, M. Melenk, eds., De Gruyter (2013)
4. Després, B.: Domain decomposition method and the Helmholtz problem. In: Mathematical and numerical aspects of wave propagation phenomena (Strasbourg, 1991), pp. 44–52. SIAM, Philadelphia, PA (1991)
5. Dolean, V., Gander, M.J., Gerardo-Giorda, L.: Optimized Schwarz methods for Maxwell's equations. SIAM J. Sci. Comput. **31**(3), 2193–2213 (2009).
6. Gander, M., Magoules, F., Nataf, F.: Optimized Schwarz methods without overlap for the Helmholz equation. SIAM J. Sci. Comp. **24**(1), 38–60 (2002)
7. Gander, M.J., Halpern, L., Magoulès, F.: An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation. Internat. J. Numer. Methods Fluids **55**(2), 163–175 (2007).
8. Hiptmair, R., Jerez-Hanckes, C.: Multiple traces boundary integral formulation for Helmholtz transmission problems. Adv. Appl. Math. **37**(1), 39–91 (2012).
9. Hsiao, G.C., Wendland, W.L.: Boundary integral equations, *Applied Mathematical Sciences*, vol. 164. Springer-Verlag, Berlin (2008).
10. Lions, J.L.: Équations différentielles opérationnelles et problèmes aux limites. Die Grundlehren der mathematischen Wissenschaften, Bd. 111. Springer-Verlag, Berlin (1961)
11. McLean, W.: Strongly Elliptic Systems and Boundary Integral Equations. Cambridge University Press, Cambridge, UK (2000)

12. Peng, Z., Lee, J.: A scalable nonoverlapping and nonconformal domain decomposition method for solving time-harmonic maxwell equations in $\mathbb{R}^3$. SIAM Journal on Scientific Computing **34**(3), 1266–1295 (2012).
13. Peng, Z., Lim, K.H., Lee, J.F.: Computations of electromagnetic wave scattering from penetrable composite targets using a surface integral equation method with multiple traces. IEEE Trans. Antennas and Propagation **61**(1), 256–270 (2013)
14. Peng, Z., Lim, K.H., Lee, J.F.: Non-conformal domain decompositions for solving large multiscale electromagnetic scattering problems. Proceedings of the IEEE **101**(2), 298–319 (2013)
15. Peng, Z., Wang, X.C., Lee, J.F.: Integral equation based domain decomposition method for solving electromagnetic wave scattering from non-penetrable objects. IEEE Trans. Antennas and Propagation **59**(9), 3328–3338 (2011)
16. Sauter, S., Schwab, C.: Boundary Element Methods, *Springer Series in Computational Mathematics*, vol. 39. Springer, Heidelberg (2010)

Recent advances in domain decomposition methods for the Stokes problem

Hyea Hyun Kim[1], Chang-Ock Lee[2], and Eun-Hee Park[3]

1 Introduction

We consider the following incompressible Stokes problem: Find $(\overrightarrow{u}, p) \in [H_0^1(\Omega)]^d \times L_0^2(\Omega)$ such that

$$\begin{aligned} -\triangle \overrightarrow{u} + \nabla p &= \overrightarrow{f}, \\ \nabla \cdot \overrightarrow{u} &= 0, \end{aligned} \tag{1}$$

where $\overrightarrow{f} \in [L^2(\Omega)]^d$ and d is the dimension of the problem domain Ω, i.e., $d = 2$ or 3. The domain Ω is assumed to be polygonal/polyhedral. The space $H_0^1(\Omega)$ is the set of square integrable functions up to first weak derivatives with zero trace on the boundary of Ω and $L_0^2(\Omega)$ is the set of square integrable functions with zero average over the domain Ω.

To find an approximate solution, a pair of inf-sup stable finite element spaces, $(\widehat{V}, \widehat{P}_0)$, is introduced such that $\widehat{V} \subset [H_0^1(\Omega)]^d$ and $\widehat{P}_0 \subset L_0^2(\Omega)$. In this work, we assume that functions in the velocity space $\widehat{V}$ are continuous. On the other hand, we can choose $\widehat{P}_0$ as discontinuous functions or as continuous functions across element boundaries. A general framework of domain decomposition algorithms will be considered for both cases of pressure functions.

There have been considerable researches on domain decomposition methods for the Stokes problem. Algorithms based on iterative substructuring methods have been developed in Marini and Quarteroni [15], Bramble and Pasciak [1], Ronquist [17], and Le Tallec and Patra [10]. Balancing Neumann-Neumann algorithms were studied by Pavarino and Widlund [16] and Goldfeld [5]. Later FETI-DP and BDDC methods were developed in the works by Li [11] and by Li and Widlund [13]. What's common in all these previous studies is that the indefinite Stokes problem is reduced to a positive definite system using the benign subspace approach. The benign subspace approach requires a compatibility condition of the velocity on the subdomain boundary as well as some primal pressure unknowns. Compared to elliptic problems, nonoverlapping domain decomposition algorithms for the Stokes problem needed careful and quite complicated construction of the coarse problem.

In recent works, more advanced algorithms were developed to address smaller and more practical coarse problems. In the works by Kim, Lee, and Park [8, 7], a coarse problem with only primal velocity unknowns was applied to the Stokes problem with a scalable condition number bound for both dual and primal forms

[1] Department of Applied Mathematics, Kyung Hee University, Yongin, Korea, e-mail: hhkim@khu.ac.kr [2] Department of Mathematical Sciences, KAIST, Daejeon, Korea, e-mail: colee@kaist.edu [3] National Institute of Mathematical Sciences, Daejeon, Korea, e-mail: eunheepark@nims.re.kr

of domain decomposition methods. In that approach a lumped preconditioner is employed. In the work by Sistek et. al. [18], extensive numerical experiments were carried out for the primal form of the Stokes problem with continuous pressure finite element functions. Similarly to [8, 7], only primal velocity unknowns are employed in their approaches. The dual form was further extended to the continuous pressure functions with a scalable condition number bound in the work by Tu and Li [12].

In the following, we introduce a general framework of domain decomposition methods for the Stokes problem and present both primal and dual domain decomposition algorithms along with estimate of their condition numbers. Throughout the paper, C is a generic positive constant independent of any mesh parameters and the number of subdomains.

2 Domain decomposition algorithms

We consider the pair of finite element spaces $(\widehat{V}, \widehat{P}_0)$. Before we proceed the construction of domain decomposition algorithms, we relax the average free condition on the pressure functions and consider the pair $(\widehat{V}, \widehat{P})$, where the pressure functions in $\widehat{P}$ are not necessarily average-free over the domain Ω. By relaxing the average-free condition on the pressure functions, the functions in $\widehat{P}$ are fully decoupled across element boundaries when discontinuous pressure functions are considered. For that case, we thus have no global pressure component but have one null component on the resulting algebraic system.

We introduce a non-overlapping subdomain partition $\{\Omega_i\}$ and decompose the function spaces into

$$V = \prod_{i=1}^{N} V_i, \quad P = \prod_{i=1}^{N} P_i,$$

where V_i and P_i are restrictions of $\widehat{V}$ and $\widehat{P}$ into Ω_i, respectively. We note that when the pressure functions in $\widehat{P}$ are discontinuous P is identical to $\widehat{P}$. In the following, we assume that the pressure functions in $\widehat{P}$ are discontinuous and we later consider the case of continuous pressure functions.

2.1 Dual formulation

In this subsection, we will present dual formulation of the Stokes problem following FETI-DP methods [3, 4] After we decouple the functions in $\widehat{V}$, we select some primal unknowns among the velocity unknowns on the subdomain boundary and enforce strong continuity on them. We use the notation $\overrightarrow{u}_\Pi$ for the primal unknowns and use the notation $\overrightarrow{u}_\Delta$ for the remaining decoupled unknowns on the subdomain interface. We call $\overrightarrow{u}_\Delta$ dual unknowns. We denote by $\overrightarrow{u}_I$ the velocity unknowns interior to each subdomains. We denote the subspaces with unknowns $\overrightarrow{u}_I$, $\overrightarrow{u}_\Delta$,

and $\overrightarrow{u}_{\Pi}$ by V_I, V_Δ, and V_Π, respectively and denote the subspace with unknowns $(\overrightarrow{u}_I, \overrightarrow{u}_\Delta, \overrightarrow{u}_\Pi)$ by $\widetilde{V}$, which has velocity unknowns that are partially coupled across the subdomain interfaces. In the dual formulation, continuity on the decoupled dual unknowns $\overrightarrow{u}_\Delta$ is enforced weakly using Lagrange multipliers $\overrightarrow{\lambda}$ and the following algebraic system will be solved:

Find $(\overrightarrow{u}_I, \overrightarrow{u}_\Delta, p, \overrightarrow{u}_\Pi, \overrightarrow{\lambda}) \in (V_I, V_\Delta, V_\Pi, P, \Lambda)$ such that

$$\begin{pmatrix} K_{II} & K_{I\Delta} & B_I^T & K_{I\Pi} & 0 \\ K_{I\Delta}^T & K_{\Delta\Delta} & B_\Delta^T & K_{\Delta\Pi} & J_\Delta^T \\ B_I & B_\Delta & 0 & B_\Pi & 0 \\ K_{I\Pi}^T & K_{I\Delta}^T & B_\Pi^T & K_{\Pi\Pi} & 0 \\ 0 & J_\Delta & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \overrightarrow{u}_I \\ \overrightarrow{u}_\Delta \\ p \\ \overrightarrow{u}_\Pi \\ \lambda \end{pmatrix} = \begin{pmatrix} \overrightarrow{f}_I \\ \overrightarrow{f}_\Delta \\ 0 \\ \overrightarrow{f}_\Pi \\ 0 \end{pmatrix} \tag{2}$$

Here Λ is the space of Lagrange multipliers λ and J_Δ is the Boolean matrix which implements weak continuity on the dual velocity unknowns $\overrightarrow{u}_\Delta$. In the above algebraic system, the unknowns $(\overrightarrow{u}_I, \overrightarrow{u}_\Delta, p)$ are fully decoupled across subdomain interfaces and can be eliminated by solving local Stokes problems and the unknowns $\overrightarrow{u}_\Pi$ then can be eliminated by solving a global coarse problem. After the elimination process, we obtain the resulting equation on λ:

$$F_d \lambda = d. \tag{3}$$

Here we stress that our formulation uses only primal velocity unknowns in contrast to the previous approaches [11, 13] which required both velocity and pressure primal unknowns satisfying a certain inf-sup stability.

The matrix F_d is symmetric and semi-positive definite on Λ. We note that F_d has null components due to fully redundant Lagrange multipliers λ_{full}

$$J_\Delta^T \lambda_{full} = 0$$

and relaxing the average-free condition on the pressure unknowns. The null component λ_{null} caused by relaxing average-free condition can be calculated by substituting $(\overrightarrow{u}_I, \overrightarrow{u}_\Delta, p, \overrightarrow{u}_\Pi, \lambda) = (0, 0, 1_p, 0, \lambda_{null})$ into (2) to obtain

$$B_\Delta^T 1_p + J_\Delta^T \lambda_{null} = 0$$

and by using $J_\Delta D_\Delta J_\Delta^T = I$, λ_{null} is given by

$$\lambda_{null} = -J_\Delta D_\Delta B_\Delta^T 1_p.$$

Here we note that D_Δ is the diagonal matrix with its entries determined by

$$D_\Delta(x) = \frac{1}{\mathscr{N}_x},$$

where $\mathscr{N}_x$ is the number of subdomains sharing the node x.

We introduce the subspace

$$\Lambda_c = \{\lambda \in \Lambda \,:\, \lambda \perp null(J_\Delta^T), \quad \lambda^T \lambda_{null} = 0\},$$

where F_d is positive definite. In our dual formulation, the equation (3) is solved on the subspace Λ_c by the preconditioned conjugate gradient method with the following lumped preconditioner

$$M_d^{-1} = J_\Delta D_\Delta K_{\Delta\Delta} D_\Delta J_\Delta^T.$$

About the performance of the proposed preconditioner, we obtain the following condition number estimate [8, 9, 6]:

Theorem 1. *In 2D when $\overrightarrow{u}_\Pi$ are selected as edge averages and in 3D when $\overrightarrow{u}_\Pi$ are selected as face averages, we obtain that*

$$\kappa(M_d^{-1}F_d) \le C\frac{H}{h}$$

and in 2D when $\overrightarrow{u}_\Pi$ are selected as values at corners we obtain that

$$\kappa(M_d^{-1}F_d) \le C\frac{H}{h}\log(1+\frac{H}{h}),$$

where H/h is the number of elements across each subdomain.

We note that the same bound was obtained for the elliptic problems with the lumped preconditioner and the same set of primal unknowns, see [14].

2.2 Primal formulation

We will now develop the primal counterpart to the dual formulation. We recall the pair of finite element spaces in the dual formulation, $(\widetilde{V}, P)$, where the velocity functions in $\widetilde{V}$ are partially coupled across the subdomain interfaces and the pressure functions in P are fully decoupled across the subdomain interfaces. We use the notations

$$\widetilde{A} := \begin{pmatrix} \widetilde{K} & \widetilde{B} \\ \widetilde{B}^T & 0 \end{pmatrix}, \quad \widetilde{J} := \begin{pmatrix} J_\Delta & 0 \end{pmatrix},$$

where $\widetilde{A}$ is the matrix obtained from the Galerkin approximation of the Stokes problem for the pair of finite element spaces $(\widetilde{V}, P)$ and $\widetilde{J}$ is the zero extension of the operator J_Δ on the pair $(\widetilde{V}, P)$. Using these notations, the dual algebraic system in (3) is written into

$$\widetilde{J}\widetilde{A}^{-1}\widetilde{J}^T\lambda = d.$$

For the primal counterpart to the dual formulation, we introduce the pair $(\widehat{V}, P)$ and obtain the algebraic equation in the primal form:

Find $(\widehat{\overrightarrow{u}}, p) \in (\widehat{V}, P)$ such that

$$\begin{pmatrix} \widehat{K} & \widehat{B} \\ \widehat{B}^T & 0 \end{pmatrix} \begin{pmatrix} \widehat{\overrightarrow{u}} \\ p \end{pmatrix} = \begin{pmatrix} \widehat{\overrightarrow{f}} \\ 0 \end{pmatrix}. \tag{4}$$

By using the extension

$$\widetilde{R} : \widehat{V} \to \widetilde{V},$$

we can express the primal form in terms of block matrices appeared in the dual form,

$$\begin{pmatrix} \widetilde{R}^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \widetilde{K} & \widetilde{B} \\ \widetilde{B}^T & 0 \end{pmatrix} \begin{pmatrix} \widetilde{R} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \widehat{\overrightarrow{u}} \\ p \end{pmatrix} = \begin{pmatrix} \widehat{\overrightarrow{f}} \\ 0 \end{pmatrix}. \tag{5}$$

We use the notation $\widehat{A}$ for the matrix in the primal form,

$$\widehat{A} = \begin{pmatrix} \widehat{K} & \widehat{B} \\ \widehat{B}^T & 0 \end{pmatrix}.$$

For the primal form, using the expression in (5) we design its preconditioner M_p^{-1} so that $M_p^{-1}\widehat{A}$ and $M_d^{-1}F_d$ have the same set of eigenvalues except zero and one. The form of the preconditioner M_p^{-1} is obtained as

$$M_p^{-1} = \begin{pmatrix} \widetilde{R}^T D & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \widetilde{K} & \widetilde{B} \\ \widetilde{B}^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} D\widetilde{R} & 0 \\ 0 & I \end{pmatrix},$$

where D is a diagonal matrix given by

$$D = \begin{pmatrix} D_\Delta & 0 \\ 0 & 0 \end{pmatrix}.$$

We note that the null component in the primal form is $(\widehat{\overrightarrow{u}}, p) = (0,1)$ and the matrix $\widehat{A}$ is indefinite. The matrix equation (4) of the primal form is solved by GMRES methods combined with the preconditioner M_p^{-1} on the subspace which is orthogonal to the null component $(\widehat{\overrightarrow{u}}, p) = (0,1)$. About the convergence of the GMRES iteration, we proved the following results:

Theorem 2. *The eigenvalues of $M_p^{-1}\widehat{A}$ and $M_d^{-1}F_d$ are the same except zero and one.*

Theorem 3. *The GMRES iteration applied to the primal form converges and its convergence is determined by ε and d, where*

$$\varepsilon = \frac{\sqrt{\lambda_{max}/\lambda_{min}} - 1}{\sqrt{\lambda_{max}/\lambda_{min}} + 1}$$

and d is purple the dimension of invariant subspaces of eigenvalues of $M_p^{-1}\widehat{A}$.

By Theorem 2 and Theorem 1, all nonzero eigenvalues of $M_p^{-1}\widehat{A}$ is real and positive. Application of M_p^{-1} to the primal form results in a two-level nonoverlapping Schwarz method, which applies an indefinite preconditioner to an indefinite problem in contrast to the dual form where a positive definite matrix is solved with the preconditioned conjugate gradient method. Under the assumption that $M_p^{-1}\widehat{A}$ is diagonalizable, the error reduction factor in the GMRES iteration is determined by

$$\|e_k\|_2 \leq C\varepsilon^k \|e_0\|_2,$$

where ε is defined in Theorem 3 and e_k is the error in the k-th iterate.

3 Application to continuous pressure functions

Algorithms in the previous section were developed for the pair $(\widehat{V},\widehat{P})$, where pressure functions in P are discontinuous across element boundaries. We will apply the algorithms to the case with continuous pressure functions. In contrast to the case with discontinuous pressure functions, we have not yet obtained the bound of eigenvalues. Instead we perform numerical experiments under various settings to see promising features of our algorithms applied to the case with continuous pressure functions.

We consider the pair $(\widehat{V},\widehat{P})$ where both velocity and pressure functions are continuous. Here we again relax the average free condition on the pressure functions as in the previous section. After we decompose the domain Ω into nonoverlapping subdomains $\{\Omega_i\}$, we obtain the decoupled velocity and pressure spaces and denote them V and P. Among those decoupled velocity unknowns on the subdomain interfaces we select some primal unknowns and enforce strong continuity on them. We denote the resulting partially coupled velocity space by $\widetilde{V}$. For the pressure functions, we can do similarly and denote the partially coupled pressure space by $\widetilde{P}$. About the pressure functions, we may not select the primal unknowns. For that case, we still use the same notation $\widetilde{P}$, which is identical to P.

After introducing these functions spaces, we obtain algebraic system in the primal form as

$$\begin{pmatrix} \widehat{K} & \widehat{B}^T \\ \widehat{B} & 0 \end{pmatrix} \begin{pmatrix} \widehat{\vec{u}} \\ \widehat{p} \end{pmatrix} = \begin{pmatrix} \widehat{\vec{f}} \\ 0 \end{pmatrix}$$

and in the dual form as

$$\begin{pmatrix} \widetilde{K} & \widetilde{B}^T & \widetilde{J}_u^T & 0 \\ \widetilde{B} & 0 & 0 & \widetilde{J}_p^T \\ \widetilde{J}_u & 0 & 0 & 0 \\ 0 & \widetilde{J}_p & 0 & 0 \end{pmatrix} \begin{pmatrix} \widetilde{\vec{u}} \\ \widetilde{p} \\ \lambda_u \\ \lambda_p \end{pmatrix} = \begin{pmatrix} \widetilde{\vec{f}} \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

where λ_u and λ_p are Lagrange multipliers for implementing weak continuity on decoupled velocity unknowns and decoupled pressure unknowns, respectively,

$$\widetilde{J}_u\widetilde{\overrightarrow{u}}=0,\quad \widetilde{J}_p\widetilde{p}=0.$$

We introduce the following notations:

$$\widetilde{A}=\begin{pmatrix}\widetilde{K} & \widetilde{B}^T\\ \widetilde{B} & 0\end{pmatrix},\quad \widetilde{J}^T=\begin{pmatrix}\widetilde{J}_u^T & 0\\ 0 & \widetilde{J}_p^T\end{pmatrix},$$

$$\widetilde{x}=\begin{pmatrix}\widetilde{u}\\ \widetilde{p}\end{pmatrix},\quad \widehat{x}=\begin{pmatrix}\widehat{u}\\ \widehat{p}\end{pmatrix},\quad \lambda=\begin{pmatrix}\lambda_u\\ \lambda_p\end{pmatrix}.$$

In addition, We introduce an extension operator

$$\widetilde{R}^T:\widehat{V}\times\widehat{P}\to\widetilde{V}\times\widetilde{P}.$$

The algebraic system in the primal form is then written as

$$\widetilde{R}\widetilde{A}\widetilde{R}^T\widehat{x}=\widehat{f}$$

and the algebraic system in the dual form after elimination process is written as

$$\widetilde{J}\widetilde{A}^{-1}\widetilde{J}^T\lambda=g.$$

For each algebraic system, we introduce preconditioners

$$M_p^{-1}=\widetilde{R}D\widetilde{A}^{-1}D\widetilde{R}^T,\quad M_d^{-1}=\widetilde{J}D\widetilde{A}D\widetilde{J}^T,$$

where D is a diagonal matrix with its entries defined similarly as before.

For the preconditioned matrices, $M_p^{-1}\widetilde{R}\widetilde{A}\widetilde{R}^T$ and $M_d^{-1}\widetilde{J}\widetilde{A}^{-1}\widetilde{J}^T$, we can prove the same result in Theorem 2. On the other hand, when the pressure functions are discontinuous the resulting matrix $\widetilde{J}\widetilde{A}^{-1}\widetilde{J}^T$ of the dual form is indefinite. Analysis of the condition number bound can not be done as in the previous section.

For the case with the continuous pressure functions, we can present the discrete problem with the following block matrices

$$\begin{pmatrix}K_{II} & B_{II}^T & K_{I\Gamma} & B_{\Gamma I}^T\\ B_{II} & 0 & B_{I\Gamma} & 0\\ K_{\Gamma I} & B_{I\Gamma}^T & K_{\Gamma\Gamma} & B_{\Gamma\Gamma}^T\\ B_{\Gamma I} & 0 & B_{\Gamma\Gamma} & 0\end{pmatrix}\begin{pmatrix}u_I\\ p_I\\ u_\Gamma\\ p_\Gamma\end{pmatrix}=\begin{pmatrix}f_I\\ 0\\ f_\Gamma\\ 0\end{pmatrix}.$$

For that case, an improvement can be done by reducing the discrete problem into the problem on the interface unknowns $(\overrightarrow{u}_\Gamma, p_\Gamma)$ and then by applying the dual and primal algorithms to the reduced interface problem. The reduction on the interface problem is called static condensation. We then observe that our dual form and primal form applied to that interface problem are similar to a FETI-DP algorithm with the Dirichlet preconditioner and a BDDC algorithm [2], respectively. Compared to the work by Li and Tu [12], our formulation employs Lagrange multipliers λ_Γ to enforce continuity on the decoupled pressure p_Γ, while p_Γ itself is treated as

Lagrange multipliers in their work. Compared to [18], our primal formulation is identical to that approach when only primal velocity unknowns are selected.

In numerical experiments, we present performance of the primal and dual forms regarding to the selection of primal unknowns and the static condensation.

4 Numerical results

We present numerical results when the algorithm for the primal form is applied to the Stokes problem discretized with $(\widehat{V},\widehat{P})$, where both the velocity and pressure functions are continuous. We refer [8, 9, 6, 7] for numerical experiments of the algorithms in Section 2, when discontinuous pressure functions are considered.

In the following numerical experiments, we consider $P_2(h)-P_1(h)$ for $2D$ problems and $Q2(h)-Q1(h)$ for $3D$ problems. The domain Ω is square/cubic and is decomposed into uniform square/cubic subdomains. In the GMRES iteration, the stop condition is when the relative residual norm is reduced by a factor of 10^6. For primal unknowns, we denote by *vc*, *ve*, and *vf* the velocity unknowns at corners, velocity averages over edges, velocity averages over faces, respectively, and we denote by *pc* the pressure unknowns at corners.

In Tables 1 and 2, for the $2D$ Stokes problem we present iteration counts depending on various sets of primal unknowns and the static condensation. As we can see, the static condensation improves a lot the iteration counts with increasing the local problem size H/h while adding more primal unknowns such as *ve* and *pc* does not give much improvement. With increasing the number of subdomains, we can observe scalability for the cases with larger set of primal unknowns, $vc+ve$ or $vc+ve+pc$.

In Tables 3 and 4, for the $3D$ Stokes problem we present iteration counts depending on various sets of primal unknowns and the static condensation. We observe similar behaviors as in the $2D$ case. The static condensation seems to be necessary to obtain good performance increasing the local problem size. About the selection of primal unknowns, in $3D$ case the additional primal unknowns *vf* improve the scalability on the number of subdomains much better than *ve* in $2D$ case. Adding *pc* does not give much improvement on the performance when increasing the number of subdomains and when increasing the local problem size.

To analyze the performance of our method depending on the set of primal unknowns and the static condensation, we plot eigenvalue distribution of the preconditioned system matrix. In Figure 1, the eigenvalue distributions in $2D$ case are presented for various sets of primal unknowns and for the cases with and without the static condensation. Among the cases without the static condensation, we observe that all eigenvalues are real and positive for the set of primal unknowns with $vc+ve+pc$. Adding *ve*, the eigenvalues become more clustered near one while adding pc does not show much improvement. About the effect of the static condensation, we see that the eigenvalues become less clustered near zero and more clustered near one. For the cases with the static condensation, we stress that the real

Table 1 2*D* Stokes problem: iteration counts depending on the set of primal unknowns and the static condensation with increasing H/h and a fixed subdomain partition $N_d = 3 \times 3$, *WOS* (without static condensation), *WS* (with static condensation)

H/h	*vc* (WOS/WS)	$vc + ve$ (WOS/WS)	$vc + ve + pc$ (WOS/WS)
2	45/27	40/25	14/14
3	58/24	46/24	22/15
4	69/25	59/21	28/16
5	78/24	66/23	35/16
6	85/25	71/23	41/17
7	93/27	88/23	47/17
8	94/26	90/22	48/18

Table 2 2*D* Stokes problem: iteration counts depending on the set of primal unknowns and the static condensation with increasing N_d and a fixed local problem size $H/h = 4$, *WOS* (without static condensation), *WS* (with static condensation)

N_d	*vc* (WOS/WS)	$vc + ve$ (WOS/WS)	$vc + ve + pc$ (WOS/WS)
3^2	69/25	59/21	28/16
4^2	92/30	71/24	29/16
5^2	108/34	70/26	30/16
6^2	117/37	69/24	30/15
8^2	138/44	67/26	30/16
10^2	146/44	69/27	30/16
12^2	147/48	67/26	30/15

Table 3 3*D* Stokes problem: iteration counts depending on the set of primal unknowns and the static condensation with increasing H/h and a fixed subdomain partition $N_d = 3^3$, *WOS* (without static condensation), *WS* (with static condensation)

H/h	*vc* (WOS/WS)	$vc + vf$ (WOS/WS)	$vc + vf + pc$ (WOS/WS)
2	16/73	56/55	40/35
3	79/75	70/55	60/40
4	98/76	77/51	73/43
5	118/74	97/52	94/43
6	134/73	120/53	117/44
7	143/75	146/54	142/45
8	149/77	171/55	167/47

Table 4 3*D* Stokes problem: iteration counts depending on the set of primal unknowns and the static condensation with increasing N_d and a fixed local problem size $H/h = 4$, *WOS* (without static condensation), *WS* (with static condensation)

N_d	vc (WOS/WS)	$vc+ve$ (WOS/WS)	$vc+ve+pc$ (WOS/WS)
3^3	79/75	70/55	60/40
4^3	109/94	77/52	67/41
6^3	203/147	79/51	68/41
8^3	227/169	76/50	65/41
9^3	301/205	93/52	87/44
10^3	298/212	93/52	87/44
12^3	288/223	93/52	87/43

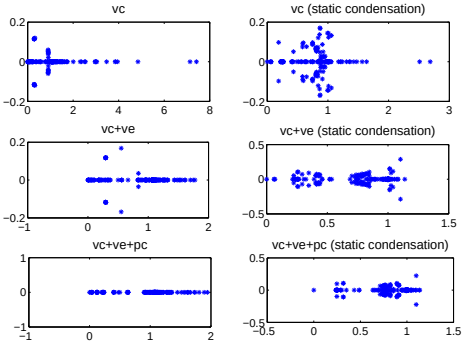


Fig. 1 2*D* Stokes problem: Eigenvalue distribution depending on the choice of primal unknowns and the static condensation, left column (without the static condensation) and right column (with the static condensation).

part of most nonzero eigenvalues are positive numbers and away from zero. In Figure 2, we plot the eigenvalue distributions for the 3*D* Stokes problem. We observe similar behaviors as in the 2*D* case. To summarize, when pressure functions in $\widehat{P}$ are continuous our algorithm with the set of primal unknowns $vc+vf$ and with the static condensation gives good performance for the 3*D* case and adding *pc* seems to be not necessary to improve the performance.

References

1. Bramble, J.H., Pasciak, J.E.: A domain decomposition technique for Stokes problems. Appl. Numer. Math. **6**(4), 251–261 (1990)
2. Dohrmann, C.R.: A preconditioner for substructuring based on constrained energy minimization. SIAM J. Sci. Comput. **25**(1), 246–258 (2003)
3. Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., Rixen, D.: FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. Internat. J. Numer. Methods Engrg. **50**(7), 1523–1544 (2001)
4. Farhat, C., Lesoinne, M., Pierson, K.: A scalable dual-primal domain decomposition method. Numer. Linear Algebra Appl. **7**(7-8), 687–714 (2000)

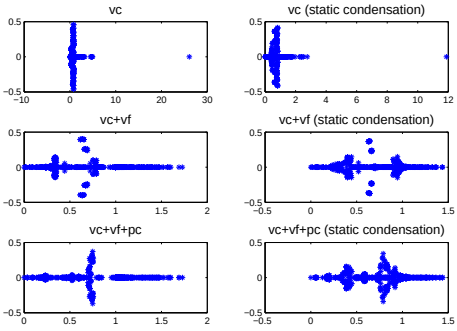


Fig. 2 3*D* Stokes problem: Eigenvalue distribution depending on the choice of primal unknowns and the static condensation, left column (without the static condensation) and right column (with the static condensation).

5. Goldfeld, P.: Balancing Neumann-Neumann preconditioners for the mixed formulation of almost-incompressible linear elasticity. ProQuest LLC, Ann Arbor, MI (2003). Thesis (Ph.D.)–New York University
6. Kim, H.H., Lee, C.O.: A FETI-DP formulation for the three-dimensional Stokes problem without primal pressure unknowns. SIAM J. Sci. Comput. **32**(6), 3301–3322 (2010)
7. Kim, H.H., Lee, C.O.: A two-level nonoverlapping Schwarz algorithm for the Stokes problem without primal pressure unknowns. Internat. J. Numer. Methods Engrg. **88**(13), 1390–1410 (2011)
8. Kim, H.H., Lee, C.O., Park, E.H.: A FETI-DP formulation for the Stokes problem without primal pressure components. SIAM J. Numer. Anal. **47**(6), 4142–4162 (2010)
9. Kim, H.H., Lee, C.O., Park, E.H.: On the selection of primal unknowns for a FETI-DP formulation of the Stokes problem in two dimensions. Comput. Math. Appl. **60**(12), 3047–3057 (2010)
10. Le Tallec, P., Patra, A.: Non-overlapping domain decomposition methods for adaptive *hp* approximations of the Stokes problem with discontinuous pressure fields. Comput. Methods Appl. Mech. Engrg. **145**(3-4), 361–379 (1997)
11. Li, J.: A dual-primal FETI method for incompressible Stokes equations. Numer. Math. **102**(2), 257–275 (2005)
12. Li, J., Tu, X.: A nonoverlapping domain decomposition method for incompressible Stokes equations with continuous pressures. SIAM J. Numer. Anal. **51**(2), 1235–1253 (2013)
13. Li, J., Widlund, O.: BDDC algorithms for incompressible Stokes equations. SIAM J. Numer. Anal. **44**(6), 2432–2455 (2006)
14. Li, J., Widlund, O.B.: On the use of inexact subdomain solvers for BDDC algorithms. Comput. Methods Appl. Mech. Engrg. **196**(8), 1415–1428 (2007)
15. Marini, L.D., Quarteroni, A.: A relaxation procedure for domain decomposition methods using finite elements. Numer. Math. **55**(5), 575–598 (1989)
16. Pavarino, L.F., Widlund, O.B.: Balancing Neumann-Neumann methods for incompressible Stokes equations. Comm. Pure Appl. Math. **55**(3), 302–335 (2002)
17. Rønquist, E.M.: Domain decomposition methods for the steady Stokes equations. In: Eleventh International Conference on Domain Decomposition Methods (London, 1998), pp. 330–340 (electronic). DDM.org, Augsburg (1999)
18. Sistek, J., Sousedik, B., Burda, P., Damasek, A., Mandel, J., Novotny, J.: Application of the parallel BDDC preconditioner to the stokes flow. Computers and Fluid **46**, 429–435 (2011)

On an Adaptive Coarse Space and on Nonlinear Domain Decomposition

Axel Klawonn[1], Martin Lanser[1], Patrick Radtke[1], and Oliver Rheinbach[2]

1 Introduction

We consider two different aspects of FETI-DP domain decomposition methods [8, 23]. In the first part, we describe an adaptive construction of coarse spaces from local eigenvalue problems for the solution of heterogeneous, e.g., multiscale, problems. This strategy of constructing a coarse space is implemented using a deflation approach. In the second part, we introduce new domain decomposition approaches for nonlinear problems. These methods are based on a decomposition of the nonlinear problem before linearization.

2 A Deflation Method

The coarse space of iterative substructuring methods such as FETI-DP or BDDC methods [8, 1, 23] can be enhanced by additional constraints using projections; see, e.g., [15]. The solution of a symmetric positive (semi-)definite system $F\lambda = d$ using the deflation method [19] also known as projector preconditioning [6], consists of the computation of λ from

$$M^{-1}(I-P)^T F\lambda = M^{-1}(I-P)^T d$$

by the conjugate gradient method using a projection of the form $P = U(U^T FU)^{-1}U^T F$ and a preconditioner M^{-1}. It is equivalent to solving $F\lambda = d$ by conjugate gradients using the symmetric preconditioner $M_{PP}^{-1} = (I-P)M^{-1}(I-P)^T$. With $\overline{\lambda} := PF^{-1}d$ the solution λ^* of the original problem is then computed as $\lambda^* = \overline{\lambda} + \lambda$. If we include the computation of $\overline{\lambda}$ into the iteration, we obtain the balancing preconditioner [17, 7] $M_{BP}^{-1} = (I-P)M^{-1}(I-P)^T + U(U^T FU)^{-1}U^T$. We then obtain the solution directly without an additional correction $\overline{\lambda}$.

For details on the deflation method or the balancing preconditioner applied to the FETI-DP or BDDC method, see [15].

[1] Mathematisches Institut, Universität zu Köln, Weyertal 86-90, 50931 Köln, Germany e-mail: {axel.klawonn}{martin.lanser}{patrick.radtke}@uni-koeln.de [2] Institut für Numerische Mathematik und Optimierung, Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg, Akademiestr. 6, 09596 Freiberg. e-mail: oliver.rheinbach@math.tu-freiberg.de

For a new coarse space for FETI-DP methods applied to almost incompressible linear elasticity in 3D implemented by deflation, see [11].

3 Coarse Spaces from Local Eigenvalue Problems

Let $\Omega \subset \mathbb{R}^2$, be a bounded polyhedral domain, let $\partial\Omega_D \subset \partial\Omega$ be a closed subset of positive measure, and $\partial\Omega_N := \partial\Omega \setminus \partial\Omega_D$ be its complement. We impose homogeneous Dirichlet and general Neumann boundary conditions on these two subsets, respectively, and introduce the Sobolev space $H^1_0(\Omega, \partial\Omega_D) := \{v \in H^1(\Omega) : v = 0 \text{ on } \partial\Omega_D\}$. We consider the piecewise linear conforming finite element approximation of the scalar diffusion problem:
Find $u \in H^1_0(\Omega, \partial\Omega_D)$, such that $a(u,v) = f(v) \ \ \forall v \in H^1_0(\Omega, \partial\Omega_D)$. Here, we use $a(u,v) := \int_\Omega \rho(x)\nabla u \cdot \nabla v \ dx$ and $f(v) := \int_\Omega fv \ dx + \int_{\partial\Omega_N} g_N v \ ds$, where g_N is the boundary data defined on $\partial\Omega_N$. We assume $\rho(x) > 0$ for $x \in \Omega$ and that ρ is piecewise constant on Ω. As a second model problem, we consider the problem of linear elasticity. For the compressible case we use the standard variational formulation to find a displacement $u \in (H^1_0(\Omega, \partial\Omega_D))^2$, such that $a(u,v) = f(v) \ \ \forall v \in (H^1_0(\Omega, \partial\Omega_D))^2$, where $a(u,v) := \int_\Omega G(x)\varepsilon(u) : \varepsilon(v) + G(x)\beta(x)\mathrm{div}(u)\mathrm{div}(v)dx$. The material parameters G and β will be expressed by $G = \frac{E}{1+\nu}$ and $\beta = \frac{\nu}{1-2\nu}$, using Young's modulus E and Poisson's ratio ν. The finite element space is denoted by V^h. We decompose Ω into N nonoverlapping subdomains Ω_i, $i = 1, \ldots, N$, where each Ω_i is the union of shape-regular and triangular finite elements with element nodes on the boundaries of neighboring subdomains matching across the interface $\Gamma := \overline{(\cup_{i=1}^N \partial\Omega_i) \setminus \partial\Omega}$. The diameter of a subdomain Ω_i is H_i or generically $H := \max_i H_i$.

Our goal is to solve multiscale, heterogenous problems with coefficient distributions as shown in Fig. 1 efficiently using the FETI-DP or BDDC method. Here, we have highly varying coefficients inside subdomains.

In the following, we will use a new approach to obtain independence of the coefficient jumps by solving local eigenvalue problems and enriching the coarse space with eigenvectors. For other approaches, designed for certain classes of coefficients; see, e.g., [14, 22]. Similar approaches have been used for Schwarz methods in [9, 5, 4]. Another approach to create adaptive coarse spaces was introduced in [18].
Let $\mathscr{E}^{ij}$ be an edge between the subdomains Ω_i and Ω_j and let $S^{(i)}_{\mathscr{E}^{ij},\rho}$ be the Schur complement that results after eliminating all variables except of the dual displacement degrees of freedom on the edge. Let $s^{(i)}_{\mathscr{E}^{ij},\rho}(u,v) := u^T S^{(i)}_{\mathscr{E}_{ij},\rho} v$ be the corresponding bilinear form and let $m_{\mathscr{E}^{ij},\rho}(u,v) := \int_{\mathscr{E}^{ij}} \rho u \cdot v \ ds$. In the case where the Poincaré constant depends on a large jump in the coefficients, we solve the following generalized eigenvalue problem on the edge: Find $u \in V^h(\mathscr{E}^{ij})$ such that

$$s^{(i)}_{\mathscr{E}^{ij},\rho}(u,v) = \mu m_{\mathscr{E}^{ij},\rho}(u,v) \qquad \forall v \in V^h(\mathscr{E}^{ij}). \tag{1}$$

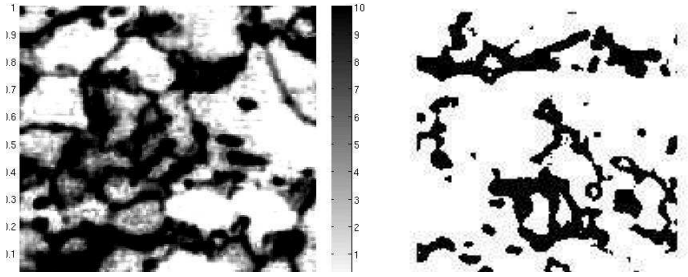


Fig. 1 Microstructures obtained from electron backscatter diffraction (EBSD/FIB). Courtesy of Prof. Dr.-Ing. Jörg Schröder, Essen, Germany, originating from a cooperation with ThyssenKrupp Steel. We have set the coefficient $E_1 = 1$ for white and $E_2 = 1e+06$ for black. An interpolated value is used for the different shades of gray. Left: gray scale image. Right: binary image. See Tab. 6 for the numerical results.

We do not need to solve this problem for all but only for the smallest eigenvalues and corresponding eigenvectors. Let the eigenvalues $0 = \mu_1 \leq \ldots \leq \mu_{n_{\mathscr{E}_{ij}}}$ be sorted in ascending order. For a given natural number $L \leq n_{\mathscr{E}_{ij}}$ and for every subdomain, we define the projection $I_L^{(l)} v := \sum_{k=1}^{L} m_{\mathscr{E}^{ij},\rho}(u_k^{(l)}, v) u_k^{(l)}$, $l = i, j$, where $u_k^{(l)}$ are the eigenvectors of (1) corresponding to the eigenvalues μ_k. In our FETI-DP algorithm and the corresponding condition number estimate, we need to force the projected jumps across the interface to be zero to obtain $I_L^{(i)} v^{(i)} = I_L^{(i)} v^{(j)}$ and $I_L^{(j)} v^{(i)} = I_L^{(j)} v^{(j)}$. Let $v_{\mathscr{E}^{ij}}^{(i)}$ be the restriction of $v^{(i)}$ to the edge $\mathscr{E}^{ij}$. To guarantee this equality, we enforce the constraint $m_{\mathscr{E}^{ij},\rho}(u_k^{(l)}, v_{\mathscr{E}^{ij}}^{(i)} - v_{\mathscr{E}^{ij}}^{(j)}) = 0$ for $k = 1, \ldots, L$ and $l = i, j$. We enrich our coarse space with the eigenvectors multiplied with the mass matrix corresponding to $m_{\mathscr{E}^{ij},\rho}$ and extended by zero on the remaining part of the interface as columns of U. We do this for each subdomain, for each edge of the subdomain, and for each eigenvector of the generalized eigenvalue problem for that edge with an eigenvalue smaller than a chosen tolerance Tol_{eig}.

The next theorem is proven in [13] under certain technical assumptions.

Theorem 1. *The condition number for our FETI-DP method satisfies*

$$\kappa(\hat{M}^{-1}F) \leq C \left(1 + \log\left(\frac{\eta}{h}\right)\right)^2 \left(1 + \frac{1}{\eta \mu_{L+1}}\right),$$

where $\hat{M}^{-1} = M_{PP}^{-1}$ or $\hat{M}^{-1} = M_{BP}^{-1}$. Here, $C > 0$ is a constant independent of H, h, and η.

Next, we present numerical results for certain exemplary coefficient distributions. We use M_{BP}^{-1} choosing M^{-1} as the Dirichlet preconditioner. We subdivide the unit square into square subdomains and consider a coefficient distribution with different numbers of channels cutting through subdomain edges; see Fig. 2. We first present our results for the scalar case followed by the results for linear elasticity with discontinuous coefficients. At the end of this section, we also present our results obtained

for the linear elastic deformation of the microstructures shown in Fig. 1. In our tables, we denote the FETI-DP algorithm using only vertices as primal constraints as "Algorithm A"; see [23, p. 170]. When the coarse space is enhanced using eigenvectors obtained from local eigenvalue problems the corresponding columns are denoted by "Adaptive". The additional constraints are implemented using deflation or balancing. They could also be implemented using a transformation of basis. Our stopping criterion is the relative reduction of the preconditioned residual by $1e-10$.

All experiments for the diffusion equation with heterogeneous coefficients inside subdomains are carried out with homogeneous Dirichlet boundary conditions on $\partial\Omega$ and a constant right hand side $f = 1/10$. For one channel for each subdomain, we have a quasi-monotone coefficient; cf. [21]. In this case, which is depicted in Fig. 2 (middle), on each interior edge, the eigenvector of the eigenvalue zero is added to the coarse space. On interior edges which do not intersect a channel with a high coefficient the resulting constraint is a standard edge average. On interior edges intersected by a channel the constraint is a weighted edge average, cf. also [14], up to a multiplicative constant. This results in eight adaptive constraints; see Tab. 1. The case of three channels results in 20 adaptive constraints.

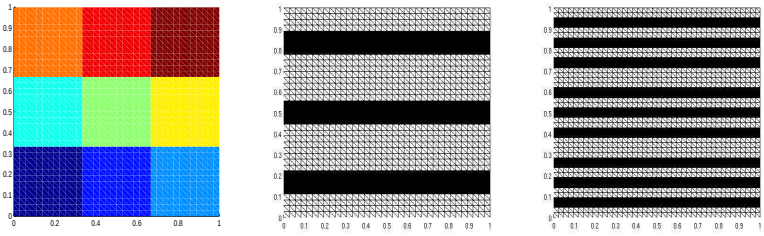


Fig. 2 Domain decomposition in nine subdomains (left). The coefficient distribution is depicted for one channel (middle) and three channels (right). Here, black corresponds to a high coefficient and white corresponds to $\rho = 1$ (middle/right).

In Tab. 2, for three channels, we see that the condition number using the enriched coarse space stays bounded if we change the contrast $\rho_2 \in \{1, \dots, 1e+06\}$. Moreover, the number of adaptive constraints approaches a limit for growing contrast.

In Tab. 3 we see that for an increasing number of subdomains and channels the condition number remains bounded. The number of adaptive constraints grows roughly in proportion to the number of subdomains and channels. Note that the adaptive algorithm with $Tol_{\text{eig}} = 1$ chooses only constraints on subdomains, where the Dirichlet boundary does not intersect the inclusions. On subdomains with Dirichlet boundary conditions that do not intersect the channels, six constraints, and on all inner subdomains, 8 constraints are chosen. Linearly dependent constraints are detected using the modified Gram-Schmidt method and removed.

Next, we test our algorithm on linear elasticity problems with certain distributions of varying coefficients inside subdomains. We impose homogeneous Dirichlet

# Channels	H/h	Algorithm A condition	# its	**Adaptive Method** condition	# its	# Adaptive constraints	Size of Γ
1	6	$9.5532e+04$	7	**1.0412**	**3**	8	84
	12	$1.1969e+05$	7	**1.1547**	**4**	8	156
	18	$1.3335e+05$	7	**1.2519**	**4**	8	228
	24	$1.4416e+05$	8	**1.3325**	**4**	8	300
	30	$1.5197e+05$	8	**1.4011**	**5**	8	372
3	14	39.2087	6	**1.0387**	**2**	20	180
	28	$1.3431e+05$	10	**1.1507**	**3**	20	348
	42	$1.3884e+05$	11	**1.2471**	**3**	20	516
	56	$1.8408e+05$	14	**1.3272**	**3**	20	684
	70	$1.9298e+05$	13	**1.3954**	**3**	20	852

Table 1 Scalar diffusion, one and three channels for each subdomain, see Fig. 2 (right). We have $\rho = 1e+06$ in the channel, and $\rho = 1$ elsewhere. The number of additional constraints is clearly determined by the structure of the heterogeneity and independent of the mesh size. $1/H = 3$. Tol_{eig}=1.

ρ_2/ρ_1	Algorithm A condition	# its	**Adaptive Method** condition	# its	# Adaptive constraints	Size of Γ
1	3.2068	5	**1.6467**	**5**	4	348
10	5.5781	7	**1.5697**	**7**	4	348
$1e+02$	19.9519	9	**1.4604**	**7**	8	348
$1e+03$	$1.5891e+02$	9	**1.1506**	**4**	20	348
$1e+04$	$1.5476e+03$	11	**1.1507**	**3**	20	348
$1e+05$	$1.5434e+04$	12	**1.1507**	**3**	20	348
$1e+06$	$1.3431e+05$	10	**1.1507**	**3**	20	348

Table 2 Scalar diffusion, three channels for each subdomain, see Fig. 2 (right). We have $\rho = \rho_2$ in the channels, and $\rho = \rho_1 = 1$ elsewhere. $H/h = 28$. The number of additional constraints is bounded for increasing contrast ρ_2/ρ_1. $1/H = 3$. Tol_{eig}=1.

$1/H$	Algorithm A condition	# its	**Adaptive Method** condition	# its	# Adaptive constraints	Size of Γ
2	1.1507	4	**1.1507**	**4**	0	114
3	$1.3431e+05$	10	**1.1507**	**3**	20	348
4	$2.3766e+05$	16	**1.1507**	**3**	44	702
5	$3.0209e+05$	45	**1.1507**	**3**	78	1176
6	$3.5451e+05$	51	**1.1507**	**3**	122	1770

Table 3 Scalar diffusion, three channels for each subdomain; see Fig. 2 (right). Increasing number of subdomains and channels. We have $\rho = 1e+06$ in the channel, and $\rho = 1$ elsewhere. $H/h = 28$. Tol_{eig}=1.

boundary conditions only on the lower edge, i.e., $y = 0$, and a constant volume force $f = (1/10, 1/10)^T$. First we consider the example above with three channels and with jumps in E instead of ρ. Tab. 4 and 5 show the numerical results for a tolerance of one for the eigenvalues. Finally, we use a coefficient distribution obtained from a steel microsection pattern with 150×150 pixels; see Fig. 1. We discretize the

problem with $H/h = 50$ and $1/H = 3$; see Tab. 6 for the numerical results, which show the effectiveness of the adaptive algorithm.

# Channels	H/h	Algorithm A condition	Algorithm A # its	**Adaptive Method** condition	**Adaptive Method** # its	# Adaptive constraints	Size of Γ
3	14	$6.8833e+05$	335	**1.1517**	**8**	123	372
	28	$9.3377e+05$	348	**1.3351**	**10**	123	708
	42	$1.0821e+06$	347	**1.4993**	**10**	123	1044

Table 4 Linear elasticity, three channels for each subdomain, see Fig. 2, with coefficient $E = 1e+06$, outside the channels $E = 1$. $Tol_{\text{eig}} = 1$. The number of additional constraints is determined by the structure of the heterogeneity and independent of the mesh size; $1/H = 3$.

E_2/E_1	Algorithm A condition	Algorithm A # its	**Adaptive Method** condition	**Adaptive Method** # its	# Adaptive constraints	Size of Γ
1	6.2497	22	**1.9264**	**12**	33	708
10	15.7940	27	**1.8460**	**12**	34	708
$1e+02$	$1.0256e+02$	39	**1.9836**	**13**	65	708
$1e+03$	$9.4413e+02$	61	**1.3398**	**9**	90	708
$1e+04$	$9.3490e+03$	117	**1.3363**	**9**	99	708
$1e+05$	$9.3373e+04$	191	**1.3352**	**9**	111	708
$1e+06$	$9.3377e+05$	348	**1.3351**	**10**	123	708

Table 5 Linear elasticity, three channels for each subdomain, see Fig. 2, $H/h = 28$. The number of additional constraints is bounded for increasing contrast E_2/E_1. $1/H = 3$, Tol_{eig}=1.

Problem	Coarse space	H/h	condition	# its	# Adaptive constraints	Size of Γ
Fig. 1 (left)	Adaptive	50	**21.6171**	**24**	114	1236
	Algorithm A	50	$> 3e+05$	> 250	0	1236
Fig. 1 (right)	Adaptive	50	**10.2617**	**22**	114	1236
	Algorithm A	50	$> 1e+06$	> 250	0	1236

Table 6 Results for linear elasticity using the coefficient distribution for the heterogenous problem from the gray scale image in Fig. 1.

4 Domain decomposition methods for nonlinear problems

The traditional domain decomposition approach to nonlinear problems can be characterized by a geometric decomposition after linearization. Here, we solve a given nonlinear, discretized problem

$$A(u) = 0 \tag{2}$$

by using a Newton-type method $u^{(k+1)} = u^{(k)} - \alpha^{(k)} \delta u^{(k)}$ with a suitable step length $\alpha^{(k)}$. In each iteration we have to solve the linearized system $DA(u^{(k)})\delta u^{(k)} = A(u^{(k)})$ which can be done by overlapping or nonoverlaping domain decomposition methods, e.g., FETI-1, FETI-DP, BDDC, or overlapping Schwarz. Such approaches are typically named NK-DD (Newton-Krylov-Domain-Decomposition), i.e., NK-FETI-DP, NK-Schwarz, etc.

Alternative approaches to the traditional DD approach can be characterized by linearization after a geometric decomposition (here denoted as DD-NK, i.e., FETI-DP-NK). Such methods can be interpreted also in the context of nonlinear preconditioning, as, e.g., performed in the ASPIN approach, see [2], which can be viewed as solving a nonlinear equation $G(A(u)) = 0$ by a Newton method instead of (2). The nonlinear preconditioner G is constructed from a nonlinear additive Schwarz (AS) method. The ASPIN approach can be classified as an AS-NK method and has been shown to be more robust and highly scalable, e.g., even for high Reynolds flow problems. Recently, the ASPIN approach has successfully been applied in nonlinear structural mechanics [12].

In this paper, we will present new approaches for nonoverlapping, nonlinear DD methods, i.e., versions of nonlinear FETI-DP methods. We will discuss two different strategies of nonlinear dual primal FETI methods, named Nonlinear-FETI-DP-1 (Linearization first) and Nonlinear-FETI-DP-2 (Elimination first).

Nonlinear, nonoverlapping domain decomposition methods have been used, in the special case of two subdomains, in multiphysics coupling, e.g., in fluid-structrure interaction; see [3]. Recently, a nonlinear FETI domain decomposition approach for nonlinear problems from elasticity was suggested by Pebrel, Rey, and Gosselet [20]. A simple linear/nonlinear strategy was used in [16] for brittle materials with strongly localized nonlinearities.

Let $\Omega_i, i = 1,\ldots,N$, be a decomposition of our domain Ω into nonoverlapping subdomains. We denote the associated local finite element spaces by W_i and the product space by $W = W_1 \times \ldots \times W_N$. We define $\widehat{W} \subset W$ as the subspace of functions from W which are continuous in all interface variables between subdomains. We consider the minimization of a global nonlinear energy function $\widehat{J}$, operating on $\widehat{W}$,

$$\hat{u} = \arg\min_{\hat{v}\in\widehat{W}} \widehat{J}(\hat{v}).$$

Using our decomposition of Ω we can build local nonlinear energy functions $J_i, i = 1,\ldots,N$, operating on W_i, and equivalently solve

$$u = \arg\min_{v\in W} \textstyle\sum_{i=1}^{N} J_i(v_i)$$

under the linear continuity constraint $Bu = 0$. Here, B is a linear jump operator, which enforces continuity in all interface variables. At this point using a variational formulation and standard dualization technique, leads us to a nonlinear saddle point problem

$$\begin{aligned} K(u) + B^T \lambda &= f \\ Bu &= 0, \end{aligned}$$

where $K(u)^T := (K_1(u_1)^T, \ldots, K_N(u_N)^T)$ and $f^T := (f_1^T, \ldots, f_N^T)$.

Using the standard FETI-DP operator R_{Π}^T, see [14] for the notation, to perform the partial assembly in the primal variables, we formulate the nonlinear FETI-DP master system

$$\begin{aligned} R_{\Pi}^T K(R_{\Pi}\tilde{u}) + B^T \lambda - \tilde{f} &= 0 \\ B\tilde{u} &= 0, \end{aligned} \tag{3}$$

where $\tilde{f} := R_{\Pi}^T f$, $\tilde{u} \in \widetilde{W}$, and the Lagrange multipliers $\lambda \in V$. Here, B enforces continuity in the dual unknowns. We can proceed in two different ways in order to solve (3). We may linearize first and then reduce the result to Lagrange multipliers (Nonlinear-FETI-DP-1), or, using the implicit function theorem, we can use nonlinear elimination and then linearization of the reduced nonlinear system (Nonlinear-FETI-DP-NK-2).

We now consider the first approach *Nonlinear-FETI-DP-1* (Linearize first). With given initial values $\tilde{u}^{(0)} \in \widetilde{W}$ and $\lambda^{(0)} \in V$, we can formulate the following Newton iteration to solve problem (3),

$$\begin{pmatrix} \tilde{u}^{(k+1)} \\ \lambda^{(k+1)} \end{pmatrix} = \begin{pmatrix} \tilde{u}^{(k)} \\ \lambda^{(k)} \end{pmatrix} - \alpha^{(k)} \begin{pmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{pmatrix},$$

with a suitable step length $\alpha^{(k)}$. In each iteration we need to solve

$$\begin{pmatrix} R_{\Pi}^T DK(R_{\Pi}\tilde{u}^{(k)})R_{\Pi} & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{pmatrix} = \begin{pmatrix} R_{\Pi}^T K(R_{\Pi}\tilde{u}^{(k)}) + B^T\lambda^{(k)} - \tilde{f} \\ B\tilde{u}^{(k)} \end{pmatrix}. \tag{4}$$

This system can be treated as in a standard FETI-DP framework, i.e., we can reduce (4) to the Lagrange multipliers. The difference to the standard NK-FETI-DP iteration can be found on the right hand side of (4). Note that, as a result of $B\delta\tilde{u}^{(k)} = B\tilde{u}^{(k)}$, jumps in the Newton update will be present only if the initial value has jumps.

In this paper, we have choosen the initial value $\lambda^{(0)} = 0$ and computed the initial value $\tilde{u}^{(0)}$ by solving the nonlinear problem

$$R_{\Pi}^T K(R_{\Pi}\tilde{u}^{(0)}) + B^T\lambda^{(0)} - \tilde{f} = 0,$$

by some Newton-type iteration. Note, that here we solve local nonlinear subdomain problems which are only coupled in the primal unknows.

Let us now consider the second approach *Nonlinear-FETI-DP-2* (Eliminate first). Instead of linearizing the nonlinear saddle point problem (3), we may perform a nonlinear elimination of the variable $\tilde{u}$ first. To simplify our notation, let us define the nonlinear operator

$$\widetilde{K}(\tilde{u}) = R_\Pi^T K(R_\Pi \tilde{u}).$$

Under sufficient assumptions the first equation of (3) can be written as

$$\tilde{u} = \widetilde{K}^{-1}(\tilde{f} - B^T\lambda), \tag{5}$$

where $\widetilde{K}^{-1}$ is the inverse map of $\widetilde{K}$. Inserting (5) into the continuity condition in (3) we obtain

$$F(\lambda) = B\widetilde{K}^{-1}(\tilde{f} - B^T\lambda) = 0. \tag{6}$$

Again we use a Newton-type iteration to solve (6), and obtain the iteration

$$\lambda^{(k+1)} = \lambda^{(k)} - \alpha^{(k)}(D_\lambda F(\lambda^{(k)}))^{-1} F(\lambda^{(k)}).$$

We can compute $D_\lambda F(\lambda)$ using the chain rule, the inverse function theorem, and (5),

$$\begin{aligned} D_\lambda F(\lambda) &= D_\lambda(B\widetilde{K}^{-1}(\tilde{f} - B^T\lambda)) = -B(D\widetilde{K}^{-1}(\tilde{f} - B^T\lambda))B^T \\ &= -B(D\widetilde{K}(\tilde{u}))^{-1}B^T = -B(R_\Pi^T(DK(R_\Pi\tilde{u})R_\Pi)^{-1}B^T. \end{aligned}$$

In each Newton step, we have to solve a nonlinear system with a FETI-DP-type matrix on the left hand side and $F(\lambda^{(k)}) = B\widetilde{K}^{-1}(\tilde{f} - B^T\lambda^{(k)})$ on the right hand side. On the right hand side nonlinear local problems have to be solved which are only coupled in the primal variables.

In contrast to a standard Newton-Krylov-FETI-DP approach, in our nonlinear FETI-DP methods weakly coupled nonlinear local problems are solved. We expect to reduce communication and to obtain a significantly improved performance especially for problems with localized nonlinearities.

Next, we introduce our nonlinear model problem and present numerical results for our two nonlinear FETI-DP approaches. Let us define the p-Laplacian for $p = 4$ as

$$\Delta_4 v = \operatorname{div}(|\nabla v|^2 \nabla v).$$

We test our algorithms for nonlinear model problems with and without localized nonlinearities. For our experiments, we consider the unit square $\Omega := [0,1] \times [0,1]$ in 2D decomposed into square subdomains $\Omega_i, i = 1, \ldots, N$. We have chosen piecewise linear triangular elements to discretize the variational formulations of (7) and (8).

First we solve the following equation for the p-Laplacian with $p = 4$ on the complete domain, i.e.,

$$\begin{aligned} \Delta_4 u &= -1 \qquad \text{in } \Omega \\ u &= 0 \qquad \text{on } \partial\Omega. \end{aligned} \tag{7}$$

In our second set of numerical experiments we consider the (linear) Laplace equation with nonlinear inclusions inside subdomains; see Fig. 3. The inclusions

are surrounded by hulls of width η. This configuration can be seen as a nonlinear analog to the problem of [10]. We denote the hull on subdomain Ω_i by $\Omega_{i,\eta}$

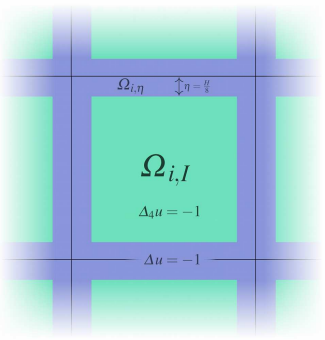


Fig. 3 Domain Ω_i with an inclusion $\Omega_{i,I}$ and $\eta = \frac{H}{8}$.

and the inclusion by $\Omega_{i,I} = \Omega_i \setminus \Omega_{i,\eta}$. Furthermore we define $\Omega_I = \bigcup_{i=1}^N \Omega_{i,I}$ and $\Omega_\eta = \bigcup_{i=1}^N \Omega_{i,\eta}$.

We then solve

$$\begin{aligned} \Delta_4 u &= -1 && \text{in } \Omega_I \\ \Delta u &= -1 && \text{in } \Omega_\eta \\ u &= 0 && \text{on } \partial\Omega. \end{aligned} \tag{8}$$

In our tests all vertices are primal and, additionally, we use primal edge constraints in our linear and nonlinear FETI-DP methods. We compare the traditional NK-FETI-DP with our nonlinear FETI-DP variants. To perform a fair comparison of the computational cost, we consider the number of Krylov space iterations and the number of linearizations separately. Each linearization includes the assembly of the local tangential matrices and their LU-decomposition. The results for problems (7) and (8) can be found in Tab. 7. The computational costs for the new methods are significantly lower for both problems, especially for the problem with local nonlinearities (p-Laplace inclusions). The number of global Krylov iterations is reduced radically and therefore, in a parallel setting, also communication.

5 Conclusion

We have presented an approach for the construction of an adaptive coarse space in FETI-DP algorithms by computing certain generalized eigenvalue problems. The method is motivated directly from the theory, i.e., a Poincaré inequality needed in the condition number estimate is now replaced by a computational bound.

		p-Laplace inclusions				p-Laplace			
N	**Solver**	**# Krylov It.**	**# Lin.**	**max. cond.**	**min. cond.**	**# Krylov It.**	**# Lin.**	**max. cond.**	**min. cond.**
4	NK-FETI-DP	33	14	1.0048	1.0001	72	18	1.1352	1.0608
	Nonlinear-FETI-DP-2	5	14	1.2813	1.0000	8	19	1.0644	1.0604
	Nonlinear-FETI-DP-1	5	15	1.2805	1.0001	12	20	1.0644	1.0604
16	NK-FETI-DP	105	15	1.4719	1.2914	164	20	1.4605	1.4107
	Nonlinear-FETI-DP-2	21	18	1.4240	1.4233	32	29	1.4208	1.4012
	Nonlinear-FETI-DP-1	28	18	1.4240	1.4233	40	24	1.4208	1.4108
64	NK-FETI-DP	164	17	1.5680	1.4264	226	22	1.5302	1.4895
	Nonlinear-FETI-DP-2	30	20	1.5255	1.5197	52	33	2.1258	1.4878
	Nonlinear-FETI-DP-1	40	20	1.5254	1.5197	52	26	2.1258	1.4850
256	NK-FETI-DP	190	19	1.5852	1.5281	268	24	1.6846	1.5394
	Nonlinear-FETI-DP-2	31	22	1.5643	1.5412	44	34	2.1523	1.5237
	Nonlinear-FETI-DP-1	42	22	1.5654	1.5406	55	28	2.1523	1.5375
1024	NK-FETI-DP	209	21	1.5786	1.4939	293	26	1.9809	1.5642
	Nonlinear-FETI-DP-2	31	24	1.5827	1.5409	45	35	2.1669	1.4921
	Nonlinear-FETI-DP-1	43	24	1.5852	1.5409	56	30	2.1669	1.5560
4096	NK-FETI-DP	215	23	1.5784	1.4972	330	28	2.5309	1.5657
	Nonlinear-FETI-DP-2	19	25	1.5768	1.5451	45	37	2.1743	1.4890
	Nonlinear-FETI-DP-1	41	26	1.5938	1.5451	45	31	2.1743	1.5588

Table 7 *p-Laplace* is described in (7) and *p-Laplace inclusions* is described in (8). For *p-Laplace inclusions*, see also Fig. 3. In both problems, $\frac{H}{h} = 16$; N is the number of subdomains; *# Krylov It.* gives the sum of all Krylov-space iterations; *# Lin.* gives the sum of all linearizations (computing local tangential matrices and their LU-decomposition); *min./max. cond* give the maximal and minimal condition number of the FETI-DP systems.

We have also presented approaches to construct nonlinear versions of the FETI-DP method. In these methods, the coarse space takes an important role since it can influence not only the convergence of the Krylov method but also that of the Newton iteration. In the future, the use of an adaptive coarse space may therefore be of special interest in this context.

Acknowledgements This work was supported (in part) by the German Research Foundation (DFG) through the Priority Programme 1648 "Software for Exascale Computing" (SPPEXA).

References

1. Dohrmann, C., *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput., 25 (2003), pp. 246–258.
2. Cai, X.C., Keyes, D.E.: Nonlinearly preconditioned inexact Newton algorithms. SIAM J. Sci. Comput. **24**(1), 183–200 (electronic) (2002). DOI 10.1137/S106482750037620X
3. Deparis, S., Discacciati, M., Fourestey, G., Quarteroni, A.: Fluid-structure algorithms based on Steklov-Poincaré operators. Comput. Methods Appl. Mech. Engrg. **195**(41-43), 5797–5812 (2006). DOI 10.1016/j.cma.2005.09.029
4. Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R., Spillane, N.: Abstract Robust Coarse Spaces for Systems of PDEs via Generalized Eigenproblems in the Overlaps. NuMa-

Report, Institute of Computational Mathematics, Johannes Kepler University Linz **7** (2011)
5. Dolean, V., Nataf, F., Scheichl, R., Spillane, N.: Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps. http://hal.archives-ouvertes.fr/docs/00/58/62/46/PDF/paper14avril.pdf (2011)
6. Dostál, Z., *Conjugate gradient method with preconditioning by projector*, Intern. J. Computer Math. **23** (1988), 315–323.
7. Dryja, M. and Widlund, O. B., *Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems*, Comm. Pure Appl. Math., **48**:2 (1995), 121–155.
8. Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., and Rixen, D., *FETI-DP: A dual-primal unified FETI method - part i: A faster alternative to the two-level FETI method*, Internat. J. Numer. Methods Engrg., **50** (2001), 1523–1544.
9. Galvis, J., Efendiev, Y.: Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces. Multiscale Model. Simul. **8**(5), 1621–1644 (2010). DOI 10.1137/100790112
10. Gippert, S., Klawonn, A., Rheinbach, O.: Analysis of FETI-DP and BDDC for linear elasticity in 3D with almost incompressible components and varying coefficients inside subdomains. SIAM J. Numer. Anal. **50**(5), 2208–2236 (2012)
11. Gippert, S., Klawonn, A., Rheinbach, O.: A Deflation Based Coarse Space in Dual-Primal FETI Methods for Almost Incompressible Elasticity Submitted to the Proceedings of the European Conference on Numerical Mathematics and Advanced Applications (ENUMATH), Lausanne, August 26-30, 2013. Springer Lecture Notes in Comput. Sci. Engrg., 8p.
12. Groß, C., Krause, R.: On the globalization of ASPIN employing trust-region control strategies - convergence analysis and numerical examples. Tech. Rep. 2011-03, Institute of Computational Science, Universita della Svizzera italiana (2011)
13. Klawonn, A., Radtke, P., Rheinbach, O.: FETI-DP methods with an adaptive coarse space. Preprint (2013), submitted for publication.
14. Klawonn, A., Rheinbach, O.: Robust FETI-DP methods for heterogeneous three dimensional elasticity problems. Comput. Methods Appl. Mech. Engrg. **196**(8), 1400–1414 (2007). DOI 10.1016/j.cma.2006.03.023
15. Klawonn, A., Rheinbach, O.: Deflation, projector preconditioning, and balancing in iterative substructuring methods: connections and new results. SIAM J. Sci. Comput. **34**(1), A459–A484 (2012). DOI 10.1137/100811118
16. Lloberas-Valls, O., Rixen, D.J., Simone, A., Sluys, L.J.: Domain decomposition techniques for the efficient modeling of brittle heterogeneous materials. Comput. Methods Appl. Mech. Engrg. **200**(13-16), 1577–1590 (2011). DOI 10.1016/j.cma.2011.01.008
17. Mandel, J., *Balancing domain decomposition*, Comm. Numer. Meth. Engrg. **9** (1993), 233–241.
18. Mandel, J., Sousedík, B.: Adaptive coarse space selection in the BDDC and the FETI-DP iterative substructuring methods: optimal face degrees of freedom. In: Domain decomposition methods in science and engineering XVI, *Lect. Notes Comput. Sci. Eng.*, vol. 55, pp. 421–428. Springer, Berlin (2007). DOI 10.1007/978-3-540-34469-8_52. URL `http://dx.doi.org/10.1007/978-3-540-34469-8_52`
19. Nicolaides, R. A., *Deflation of conjugate gradients with applications to boundary value problems*, SIAM J. Numer. Anal., **24:2** (1987), 355–365.
20. Pebrel, J., Rey, C., Gosselet, P.: A nonlinear dual-domain decomposition method: Application to structural problems with damage. Inter. J. Multiscal Comp. Eng. **6**(3), 251–262 (2008)
21. Pechstein, C., Scheichl, R.: Weighted Poincaré inequalities. NuMa-Report, Institute of Computational Mathematics, Johannes Kepler University Linz **10** (2010)
22. Pechstein, C., Scheichl, R.: Analysis of FETI methods for multiscale PDEs. Part II: interface variation. Numer. Math. **118**(3), 485–529 (2011). DOI 10.1007/s00211-011-0359-2
23. Toselli, A. and Widlund, O. B., *Domain decomposition methods-algorithms and theory*, vol. 34, Springer, 2004.

On Iterative Substructuring Methods for Multiscale Problems

Clemens Pechstein[1]

1 Introduction

Model Problem Let $\Omega \subset \mathbb{R}^2$ or $\mathbb{R}^3$ be a Lipschitz polytope with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, where $\Gamma_D \cap \Gamma_N = \emptyset$. We are interested in finding $u_h \in V_D^h(\Omega)$ such that

$$\int_\Omega \alpha \nabla u_h \cdot \nabla v_h \, dx \;=\; \langle f, v_h \rangle \qquad \forall u_h \in V_D^h(\Omega). \tag{1}$$

Above, $V_D^h(\Omega)$ denotes the finite element space of continuous and piecewise linear functions with respect to a mesh $\mathscr{T}^h(\Omega)$ that vanish on the Dirichlet boundary Γ_D. The functional $f \in V_D^h(\Omega)^*$ is assumed to be composed of a volume integral over Ω and a surface integral over Γ_N.

The diffusion coefficient $\alpha \in L^\infty(\Omega)$ is assumed to be uniformly positive, i.e., $\operatorname{ess.inf}_{x\in\Omega} \alpha(x) > 0$. We allow α to vary by several orders of magnitude in an unstructured way throughout the domain Ω. In particular, we allow α to be discontinuous and exhibit large jumps (high contrast). If the jumps occur at a scale $\eta \ll \operatorname{diam}(\Omega)$, one speaks of a *multiscale problem* (cf. e.g., [1]).

Problem (1) is equivalent to the linear system

$$K_{h,\alpha}\, \underline{u}_h \;=\; \underline{f}_h\,, \tag{2}$$

where the stiffness matrix $K_{h,\alpha}$ and load vector $\underline{f}_h$ are defined with respect to the standard nodal basis of $V_D^h(\Omega)$. For a quasi-uniform mesh, one easily shows that

$$\kappa(K_{h,\alpha}) \le C \frac{\operatorname{ess.sup}_{x\in\Omega} \alpha(x)}{\operatorname{ess.inf}_{x\in\Omega} \alpha(x)}\, h^{-2}\,.$$

Although in many cases, this might be a pessimistic bound, it is sharp in general. Consequently, an ideal preconditioner for $K_{h,\alpha}$ should be robust in (i) the contrast in α, (ii) the mesh size h, (iii) the scale η at which the coefficient varies, where here we may assume that $h \le \eta \le \operatorname{diam}(\Omega)$.

Spectral Properties and the Weighted Poincaré Inequality To get an idea, how difficult it is to precondition System (2), we display the entire *spectrum* of $K_{h,\alpha}$ for the pure Neumann problem ($\Gamma_D = \emptyset$) on the unit square $\Omega = (0,1)^2$ and for three coefficient distributions α (see the top row of Fig. 1). The smallest eigenvalue of $K_{h,\alpha}$ is always zero and not shown in the following plots.

[1]Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Altenberger Str. 69, 4040 Linz, Austria e-mail: clemens.pechstein@oeaw.ac.at

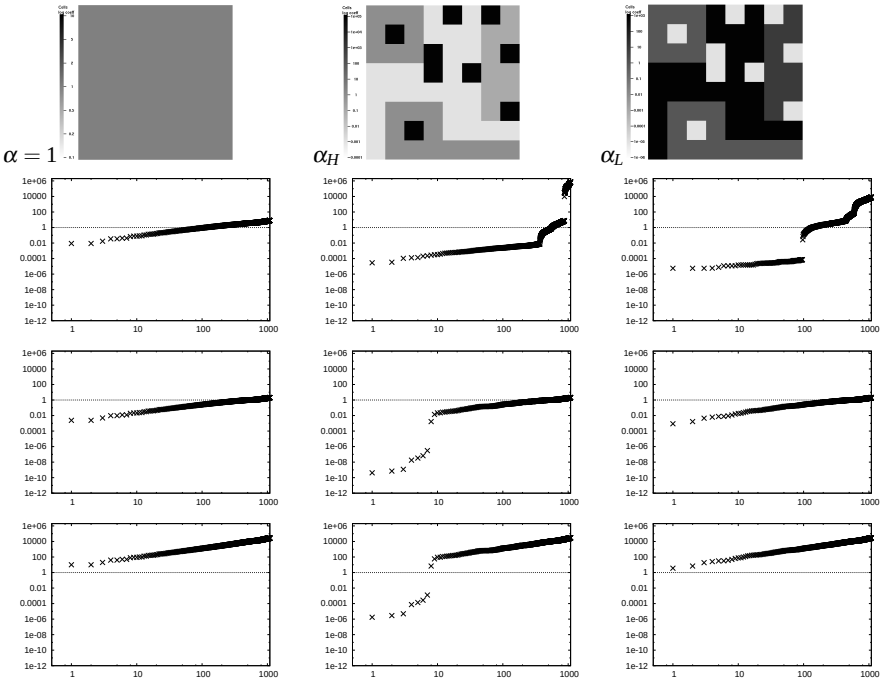


Fig. 1 Top row: three coefficient distributions α. Second row: spectra $\sigma(K_{h,\alpha})$ corresponding to the three distributions. Third row: $\sigma(\mathrm{diag}(K_{h,\alpha})^{-1}K_{h,\alpha})$. Bottom row: $\sigma(M_{h,\alpha}^{-1}K_{h,\alpha})$. In each case structured mesh with mesh size $h = 1/32$. The contrast for $\alpha_H = \alpha_L^{-1}$ is 10^8.

The second row of Fig. 1 displays $\sigma(K_{h,\alpha})$. We see that compared to the reference coefficient $\alpha = 1$, the spectrum is distorted in the two other cases α_H, α_L.

In the third and fourth row, we change the point of view, and display the spectrum of $\mathrm{diag}(K_{h,\alpha})^{-1}K_{h,\alpha}$ and of $M_{h,\alpha}^{-1}K_{h,\alpha}$, where $M_{h,\alpha}$ denotes the weighted mass matrix corresponding to the inner product $(v,w)_{L^2(\Omega),\alpha} := \int_\Omega \alpha\, v w\, dx$. On a quasi-uniform mesh, one can easily show that $\mathrm{diag}(K_{h,\alpha})$ and $h^{-2}M_{h,\alpha}$ are spectrally equivalent with uniform constants. For this reason, the spectra in the third and fourth row differ mainly by a simple shift. For coefficient α_H, with 8 inclusions of large values (plotted in black), we obtain 7 additional small eigenvalues compared to the reference coefficient. This fact has been theoretically shown by Graham & Hagger [10].

For coefficient α_L, with 8 inclusions of small values (plotted in light grey), the spectra are essentially the same as for the reference coefficient. The theoretical explanation of this fact is the so-called *weighted Poincaré inequality* [17].

Definition 1. Let $\{D_i\}$ be a finite partition of Ω into polytopes, let α be piecewise constant w.r.t. $\{D_i\}$ with value α_i on D_i, and let ℓ^* be an index such that $\alpha_{\ell^*} = \max_i \alpha_i$. Then α is called *quasi-monotone* on Ω iff for each i we can find a path $D_{\ell_1} \cup D_{\ell_2} \cup \ldots \cup D_{\ell_n}$ of subregions connected through proper faces with $\ell_1 = i$, $\ell_n = \ell^*$ such that $\alpha_{\ell_1} \le \alpha_{\ell_2} \le \ldots \le \alpha_{\ell_n}$.

Def. 1 is independent of the choice of ℓ^*: if α attains its maximum in more than one subregion, then α is either not quasi-monotone, or all the maximum subregions are connected. In our example, α_L is quasi-monotone, whereas α_H is not.

Theorem 1. *If α (as in Def. 1) is quasi-monotone on Ω, then there exists a constant $C_{P,\alpha}(\Omega)$ independent of the* values α_i *and of* $\operatorname{diam}(\Omega)$ *such that*

$$\inf_{c\in\mathbb{R}} \|u-c\|_{L^2(\Omega),\alpha} \le C_{P,\alpha}(\Omega)\operatorname{diam}(\Omega)\,|u|_{H^1(\Omega),\alpha} \qquad \forall u \in H^1(\Omega),$$

where $\|v\|^2_{L^2(\Omega),\alpha} := \int_\Omega \alpha\, v^2\,dx$ *and* $|v|_{H^1(\Omega),\alpha} := \int_\Omega \alpha\,|\nabla v|^2\,dx$.

For the *geometrical* dependence of $C_{P,\alpha}(\Omega)$ on the partition $\{D_i\}$ (in our previous example, the scale η), we refer to [17]. The infimum on the left hand side is attained at the weighted average $c = \overline{u}^{\Omega,\alpha} := \int_\Omega \alpha\, u\,dx / \int_\Omega \alpha\,dx$. Due to the fact that the coefficient α_L in Fig. 1 is *quasi-monotone*, $\lambda_2(M^{-1}_{h,\alpha}K_{h,\alpha}) \ge C_{P,\alpha}(\Omega)^{-2}\operatorname{diam}(\Omega)^{-2}$ and thus bounded from below independently of the contrast in α_L.

Related Preconditioners The simple examples in Fig. 1 show that it is not necessarily contrast alone, which makes preconditioning difficult, but a special *kind* of contrast. The fact that a *small* number of large inclusions leads to essentially well-conditioned problems has, e.g., been exploited in [22]. Overlapping Schwarz theory is given in [11] for coefficients of type α_H, and in [7, 18] for *locally* quasi-monotone coefficients. Robustness theory of FETI methods for locally quasi-monotone coefficients has been developed in [15, 16, 14, 13]. Achieving *robustness* in the general case requires a good coarse space (either for overlapping Schwarz or FETI). Spectral techniques, in particular solving local generalized eigenvalue problems to *compute* coarse basis functions, have come up in [8, 5, 19] (see also the references therein). Very recently, this approach has been even carried over to FETI methods by Spillane and Rixen [20]; see also Axel Klawonn's DD21 talk and proceedings contribution. Although the spectral approaches above guarantee *robust* preconditioners, the dimension of the coarse space may be large, therefore making the preconditioner inefficient. For analyzing the coarse space dimension, tools like the weighted Poincaré inequality are quite useful, cf. [5].

Outline In this paper, we shall

(i) review the available theoretical results of FETI methods for coefficients that are—on each subdomain (or a part of it)—quasi-monotone (i.e., of type α_L),
(ii) present novel theoretical robustness results of FETI methods for coefficients which result from a large number of inclusions with *large* values (i.e., of type α_H far from quasi-monotone). In particular, we allow the inclusions to cut through or touch certain interfaces of the (non-overlapping) domain decomposition.

In both cases, the coarse space is the usual space of constants in each subdomain. After fixing some notation in Sect. 2, we present our review (i) in Sect. 3. Sect. 4 deals with technical tools needed for the novel theory of (ii), which is contained in Sect. 5. In the end, we draw some conclusions.

2 FETI and TFETI

FETI Basics We briefly introduce classical and total FETI; for details see e.g., [21, 13]. The domain Ω is decomposed into non-overlapping subdomains $\{\Omega_i\}_{i=1}^s$, resolved by the fine mesh $\mathscr{T}^h(\Omega)$. The *interface* is defined by $\Gamma := \bigcup_{i\neq j=1}^s (\partial\Omega_i \cap \partial\Omega_j) \setminus \Gamma_D$. Let K_i denote the "Neumann" stiffness matrix corresponding to the local bilinear form $\int_{\Omega_i} \alpha \nabla u \cdot \nabla v dx$, and let S_i be the Schur complement of K_i after eliminating the interior degrees of freedom and those corresponding to non-coupling nodes on the Neumann boundary. In the *classical* variant of FETI [6], the corresponding local spaces are chosen to be

$$W_i := \{v \in V^h(\partial\Omega_i \setminus \Gamma_N) : v_{|\Gamma_D} = 0\}.$$

In the case of the *total FETI* (TFETI) method [4], the Dirichlet boundary conditions are not included into K_i, and correspondingly $W_i := V^h(\partial\Omega_i \setminus \Gamma_N)$. We set $W := \prod_{i=1}^s W_i$ and $S := \mathrm{diag}(S_i)_{i=1}^s$. Let R be a block-diagonal full-rank matrix such that $\ker(S) = \mathrm{range}(R)$, and let $B : W \to U$ be a jump operator such that $\ker(B) = \widehat{W}$, where $\widehat{W} \subset W$ is the space of functions being continuous across Γ and fulfilling the homogeneous Dirichlet boundary conditions. The rows of $Bu = 0$ are formed by all (fully redundant) constraints $u_i(x^h) - u_j(x^h) = 0$ for $x^h \in \partial\Omega_i \cap \partial\Omega_j \setminus \Gamma_D$. In TFETI, there are further local constraints of the form $u_i(x^h) = 0$ for $x^h \in \partial\Omega_i \cap \Gamma_D$. Finally, System (2) is reformulated as $\begin{bmatrix} S & B^\top \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$, where f contains the reduced local load vectors, and further reformulated by

$$\text{find } \widetilde{\lambda} \in \mathrm{range}(P): \quad P^\top F \widetilde{\lambda} \;=\; \widetilde{d} := P^\top B S^\dagger (f - B^\top \lambda_0), \tag{3}$$

where $S^\dagger$ is a pseudo-inverse of S, $F := BS^\dagger B^\top$, $P := I - QG(G^\top QG)^{-1}G^\top$, $G := BR$, $\lambda_0 = QG(G^\top QG)^{-1}R^\top f$, and Q is yet to be specified. The solution u can be recovered easily from $\lambda = \lambda_0 + \widetilde{\lambda}$ by using $S^\dagger$ and $(G^\top QG)^{-1}$.

Scaled Dirichlet Preconditioner For each subdomain index j and each degree of freedom (i.e., node) $x^h \in \partial\Omega_j \cap \Gamma$, we fix a weight $\rho_j(x^h) > 0$ and define

$$\delta_j^\dagger(x^h) := \frac{\rho_j(x^h)^\gamma}{\sum_{k\in\mathscr{N}_{x^h}} \rho_k(x^h)^\gamma} \;\in [0,1], \qquad \sum_{j\in\mathscr{N}_{x^h}} \delta_j^\dagger(x^h) = 1.$$

Above, $\mathscr{N}_{x^h}$ is the set of subdomain indices sharing node x^h and $\gamma \in [1/2, \infty]$ (the limit $\gamma \to \infty$ has to be carried out properly, cf. [13, Rem. 2.27]). We stress that in the presence of jumps in α, the choice of the weights $\rho_j(x^h)$ (or the scalings $\delta_j^\dagger(x^h)$) is highly important for the robustness of the Dirichlet preconditioner and will be discussed further below. Let us note that for any choice $\rho_j(x^h)$ above and any exponent $\gamma \in [1/2, \infty]$, we have the elementary inequality

$$\rho_i(x^h)\, \delta_j^\dagger(x^h)^2 \;\le\; \min(\rho_i(x^h), \rho_j(x^h)) \qquad \forall i, j \in \mathscr{N}_{x^h}. \tag{4}$$

The weighted jump operator B_D is defined similarly to B, but each row of $B_D w = 0$ is of the form $\delta_j^\dagger(x^h)\, w_i(x^h) - \delta_i^\dagger(x^h)\, w_j(x^h) = 0$ for $x^h \in \partial\Omega_i \cap \partial\Omega_j \setminus \Gamma_D$. In TFETI, there are further rows of the form $w_i(x^h) = 0$ for $x^h \in \partial\Omega_i \cap \Gamma_D$. The preconditioned FETI system now reads

$$\text{find } \widetilde{\lambda} \in \text{range}(P): \quad PM^{-1}P^\top F\,\widetilde{\lambda} \;=\; PM^{-1}\,\widetilde{d}, \tag{5}$$

where $M^{-1} := B_D S B_D^\top$. Since $P^\top F$ is SPD on range(P) up to ker$(B^\top)$, this system can be solved by CG. Hence, one is interested in a bound on the condition number $\kappa_{\text{FETI}} := \kappa(PM^{-1}P^\top F_{|\text{range}(P)/\ker(B^\top)})$. In the sequel, we set $Q = M^{-1}$. To avoid complications, we exclude the case of TFETI with $\Gamma_D = \partial\Omega$, and the case $\gamma = \infty$; otherwise $GM^{-1}G^\top$ may be singular. As the analysis in [21], [13, Chap. 2] shows, the estimate

$$|P_D\, w|_S^2 \;\le\; \mu\, |w|_S^2 \qquad \forall w \in W^\perp, \tag{6}$$

implies $\kappa_{\text{FETI}} \le 4\mu$. Above, $P_D := B_D^\top B$ is a *projection* (due to the partition of unity property of $\delta_j^\dagger$), $W^\perp = \prod_{i=1}^s W_i^\perp$, and each $W_i^\perp \subset W_i$ is any complementary subspace such that the sum $W_i = \ker(S_i) + W_i^\perp$ is direct. Note that the same estimate implies a bound of the related balancing Neumann-Neumann (BDD) method.

Choice of Weights Table 1 shows several choices for the weights $\rho_j(x^h)$. In each row, we display a *theoretical* choice, which has been used in certain analyses, and then a *practical* choice, which tries to mimic the theoretical one. Choices (a)–(c) in Table 1 are not suitable for coefficients with jumps (see column *problems*). The theoretical choice (d) will be used in the analyses below and leads to "good" condition number bounds under suitable assumptions; however, it is practically infeasible. Under suitable assumptions on the variation of α, the practical choice (d) can be shown to be essentially equivalent to the theoretical one, if one sets $\gamma = \infty$. "Good" means that the bounds are robust with respect to contrast in α. However, they depend on the spatial scale η of the coefficient variation.

$\rho_j(x^h)$	*theoretical*	*practical*	*problems*
(a)	1	1 (multiplicity scaling)	jumps across interfaces
(b)	$\alpha_{\Omega_j}^{\max}$	$\|K_j^{\text{diag}}\|_{\ell^\infty}$	jumps within subdomains
(c)	$\max_{\tau\subset\Omega_j : x^h\in\overline{\tau}} \alpha_{\|\tau}$	$K_j^{\text{diag}}(x^h)$ (stiffness scaling)	oscillating coefficients, unstructured meshes
(d)	$\max_{Y_j^{(k)} : x^h\in\overline{Y}_j^{(k)}} \alpha_{Y_j^{(k)}}^{\max}$	$\begin{cases} 1 & \text{if } K_j^{\text{diag}}(x^h) \simeq \max_{k\in\mathcal{N}_{x^h}} K_k^{\text{diag}}(x^h) \\ 0 & \text{else} \end{cases}$	small geometric scale η

Table 1 Various choices for the weights $\rho_j(x^h)$. Here, K_j^{diag} denotes the diagonal of K_j, $\|\cdot\|_{\ell^\infty}$ the maximum norm, $K_j^{\text{diag}}(x^h)$ the diagonal entry of K_j corresponding to node x^h, and $\{Y_j^{(k)}\}_k$ is a partition of a neighborhood of $\partial\Omega_j \cap \Gamma$, as coarse as possible, such that α is constant or only mildly varying in each subregion $Y_j^{(k)}$, cf. [13, Sect. 3.3].

Remark 1. A further choice, named *Schur scaling*, has been suggested in [3], see also [2]. There, for each subdomain vertex/edge/face $\mathscr{G}$, the *scalar* values $\delta_j^\dagger(x^h)$ for $x^h \in \mathscr{G}$ are replaced by the *matrix* $(\sum_{k\in\mathscr{N}_\mathscr{G}} S_{k,\mathscr{G}\mathscr{G}})^{-1} S_{j,\mathscr{G}\mathscr{G}}$, where $S_{k,\mathscr{G}\mathscr{G}}$ denotes the restriction of S_k to the nodes on the subdomain vertex/edge/face $\mathscr{G}$. This choice is the only known (practical) candidate that could allow for robustness also with respect to the spatial scale η, but its analysis is still under development, cf. [2]. Nevertheless, it has been successfully analyzed in the context of BDDC methods for the eddy current problem $\overrightarrow{curl}(\alpha \overrightarrow{curl}\, \overrightarrow{u}) + \beta\, \overrightarrow{u} = \overrightarrow{f}$, where $\alpha, \beta > 0$ are constant in each subdomain [3].

3 Robustness Results for Locally Quasi-monotone Coefficients

In this section, we review robustness results of TFETI, developed originally in [15, 16] and further refined in [13, Chap. 3]. Because of space limitation, we do not list the full set of assumptions, but refer to [13, Sect. 3.3.1, Sect. 3.5]. The essential assumption is that α is piecewise constant with respect to a shape-regular mesh $\mathscr{T}^\eta(\Omega)$, at least in the neighborhood of the interface Γ and the Dirichlet boundary Γ_D, and that this mesh resolves $\Gamma \cup \Gamma_D$. For simplicity of the presentation, we assume further that each subdomain Ω_i is the union of a few elements of a coarse mesh $\mathscr{T}^H(\Omega)$, and that the three meshes $\mathscr{T}^h(\Omega)$, $\mathscr{T}^\eta(\Omega)$, and $\mathscr{T}^H(\Omega)$ are nested, shape-regular, and globally quasi-uniform with mesh parameters $h \le \eta \le H$.

All the following results hold for the TFETI method as defined in Sect. 2 with the theoretical choice (d) for $\rho_j(x^h)$ and with $Q = M^{-1}$, where the regions $Y_j^{(k)}$ are unions of a few elements from $\mathscr{T}^\eta(\Omega)$. The general bound reads

$$\kappa_{\text{FETI}} \;\le\; C\left(\frac{H}{\eta}\right)^\beta (1+\log(\eta/h))^2\,, \tag{7}$$

where C is independent of H, η, h, and α. The exponent β is specified below in each particular case.

Definition 2. For each subdomain index i, the *boundary layer* $\Omega_{i,\eta}$ is the union of those elements from $\mathscr{T}^\eta(\Omega)$ that lie in Ω_i and touch $\Gamma \cup \Gamma_D$.

The following theorem is essentially [13, Thm. 3.64] and shows that contrast in the interior of subdomains is taken care of by TFETI (in form of the subdomain solves), except that the geometrical scale shows up in the condition number bound. The original result on classical FETI can be found in [15, Thm. 3.3].

Theorem 2 (Constant Coefficients in the Boundary Layers). *If α is constant in each boundary layer $\Omega_{i,\eta}$, $i = 1,\dots,s$, then (7) holds with $\beta = 2$. The exponent $\beta = 2$ is sharp in general. If the values of α in $\Omega_i \setminus \Omega_{i,\eta}$ do not fall below the constant value in $\Omega_{i,\eta}$ for each $i = 1,\dots,s$, then (7) holds with $\beta = 1$.*

The next theorem (cf. [13, Sect. 3.5.2]) extends the above result to coefficients that are quasi-monotone in each boundary layer.

Theorem 3 (Quasi-monotone coefficients in the Boundary Layers). *If α is quasi-monotone in each boundary layer $\Omega_{i,\eta}$, $i = 1,\dots,s$, then (7) holds with $\beta = 2$ if $d = 2$ and $\beta = 4$ if $d = 3$. Under suitable additional assumptions on α in $\Omega_{i,\eta}$, one can achieve $\beta = 2$ for $d = 3$ as well.*

In many cases, quasi-monotonicity may not hold in each boundary layer, but in a certain sense on a larger domain. The following theorem summarizes essentially [13, Sect. 3.5.3]. We note that the concept of an *artificial coefficient* in the context of FETI goes back to [16].

Theorem 4 (Quasi-monotone Artificial Coefficients). *If for each $i = 1,\dots,s$ there exists an auxiliary domain Λ_i with $\Omega_{i,\eta} \subset \Lambda_i \subset \Omega_i$ and an* artificial coefficient α^{art} *such that*

$$\begin{aligned} &\alpha^{\mathrm{art}} = \alpha \quad \text{in } \Omega_{i,\eta}, \\ &\alpha^{\mathrm{art}} \le \alpha \quad \text{in } \Lambda_i \setminus \Omega_{i,\eta}, \\ &\alpha^{\mathrm{art}} \text{ quasi-monotone on } \Lambda_i, \end{aligned}$$

then (7) holds with C independent of α and α^{art}. The exponent β depends on Λ_i and α^{art}. If $\Lambda_i = \Omega_i$ then $\beta \le 2d$. Under additional assumptions on α^{art}, one can achieve, e.g., $\beta \le d+1$.

Remark 2. The proofs of Thm. 3 and Thm. 4 make heavy use of the weighted Poincaré inequality (Thm. 1). We note that Thm. 3 and Thm. 4 can be generalized to so-called type-*m* quasi-monotonicity (see [17]). Also, all the results of this section can be generalized to (i) coefficients that vary mildly in each element of $\mathscr{T}^\eta(\Omega)$ in the neighborhood of $\Gamma \cup \Gamma_D$, (ii) to a certain extent to suitable diagonal choices of the matrix Q, and (iii) under suitable conditions to classical FETI. However, we do not present these results here but refer to [13, Chap. 3] and [15, 16] for the full theory.

4 Technical Tools

In this section, we present two technical tools needed for Sect. 5. The first tools is an extension operator on so-called *quasi-mirrors*.

Definition 3. Let $D_1, D_2 \subset \mathbb{R}^d$ be two disjoint Lipschitz domains sharing a $(d-1)$-dimensional manifold Γ. For $i = 1,2$ let D_{ia} and D_{ib} be open and disjoint Lipschitz domains such that $\overline{D}_i = \overline{D}_{ia} \cup \overline{D}_{ib}$. We say that (D_{2a}, D_{2b}) is a *quasi-mirror* of (D_{1a}, D_{1b}) iff there exists a continuous and piecewise C^1 bijection ϕ with $\|\nabla\phi\|_{L^\infty}$ and $\|\nabla\phi^{-1}\|_{L^\infty}$ bounded, such that D_{ia}, D_{ib}, Γ are mapped to $\hat{D}_{ia}$, $\hat{D}_{ib}$, $\hat{\Gamma}$, respectively, where $\hat{\Gamma}$ lies in the hyperplane $x_d = 0$ and $\hat{D}_{2a}$, $\hat{D}_{2b}$ are the reflections through that hyperplane of $\hat{D}_{1a}$, $\hat{D}_{1b}$, respectively (for an illustration see Fig. 2).

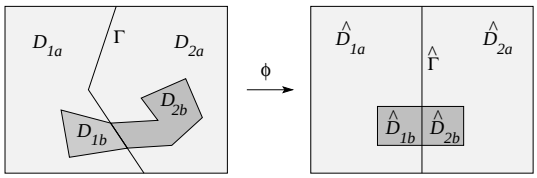


Fig. 2 Illustration of Def. 3: a quasi-mirror in 2D.

Lemma 1. *Let (D_{2a}, D_{2b}) be a quasi-mirror of (D_{1a}, D_{1b}) as in Def. 3. Then there exists a linear operator $E : H^1(D_1) \to H^1(D_2)$ such that for all $v \in H^1(D_1)$, we have $(Ev)_{|\Gamma} = v_{|\Gamma}$ and*

$$|Ev|_{H^1(D_{2a})} \le C\,|v|_{H^1(D_{1a})}, \qquad |Ev|_{H^1(D_{2b})} \le C\,|v|_{H^1(D_{1b})},$$
$$\|Ev\|_{L^2(D_{2a})} \le C\,\|v\|_{L^2(D_{1a})}, \qquad \|Ev\|_{L^2(D_{2b})} \le C\,\|v\|_{L^2(D_{1b})}.$$

The constant C is dimensionless, but depends on the transformation ϕ from Def. 3.

The proof of the above and the next lemma can be found in [12, Sect. 4]. Our second tool is a special Scott-Zhang quasi-interpolation operator.

Lemma 2. *Let the domain D be composed from two disjoint Lipschitz regions $\overline{D} = \overline{D}_1 \cup \overline{D}_2$ with interface $\Gamma = \partial D_1 \cap \partial D_2$, and let $\Sigma \subset \partial D$ be non-trivial. Let $\mathscr{T}^h(D)$ be a shape-regular mesh resolving Γ and Σ, and let $V^h(D)$ denote the corresponding space of continuous and piecewise linear finite element functions. Then there exists a projection operator $\Pi_h : H^1(D) \to V^h(D)$ such that (i) for any $v \in H^1(D)$ that is piecewise linear on Γ and Σ, $(\Pi_h v)_{\Gamma\cup\Sigma} = v_{|\Gamma\cup\Sigma}$ and (ii) for all $v \in H^1(D)$,*

$$|\Pi_h v|_{H^1(D_i)} \le C\,|v|_{H^1(D_i)}, \qquad \|\Pi_h v\|_{L^2(D_i)} \le C\,\|v\|_{L^2(D_i)}, \qquad \text{for } i = 1, 2,$$

where the constant C only depends on the shape-regularity of the mesh.

5 Novel Robustness Results for Inclusions

For this section, we adopt again the notations of Sect. 2 and 3. However, we restrict ourselves to coefficients $\alpha \in L^\infty(\Omega)$, given by

$$\alpha(x) \;=\; \begin{cases} \alpha_k & \text{if } x \in D_k \text{ for some } k = 1, \dots, n_H\,, \\ \alpha_L & \text{else,} \end{cases} \tag{8}$$

where $\alpha_k \ge \alpha_L$ are constants and the regions $\overline{D}_k \subset \overline{\Omega}$ are pairwise disjoint (disconnected) Lipschitz polytopes that are contractible (i.e., topologically isomorphic to the ball). Furthermore, we assume that the subdomains Ω_i as well as the inclusion regions D_k are resolved by a global mesh $\mathscr{T}^\eta(\Omega)$. For the sake of simplicity let $\mathscr{T}^h(\Omega)$ and $\mathscr{T}^\eta(\Omega)$ be nested, shape-regular, and quasi-uniform with mesh sizes h and η, respectively ($h \le \eta$). Our main assumption concerns the location of the inclusion regions D_k relative to the interface.

Assumption A1. Each region D_k, $k = 1, \dots, n_H$, is either
(a) an *interior inclusion*: $D_k \subset\subset \Omega_i$ for some index i,
(b) a *docking inclusion*: there is a unique index i with $D_k \subset \Omega_i$ and $\overline{D}_k \cap \partial\Omega_i \neq \emptyset$, or
(c) a *(proper) face inclusion*: there exists a subdomain face $\mathscr{F}_{ij}$ (shared by only two subdomains Ω_i, Ω_j) such that

- $\overline{D}_k \cap \Gamma \subset\subset \mathscr{F}_{ij}$,
- $\partial(D_k \cap \Omega_i) \cap \mathscr{F}_{ij} = \partial(D_k \cap \Omega_j) \cap \mathscr{F}_{ij}$,
- $\overline{D}_k \cap \Gamma$ is simply connected,
- the neighborhood $\mathscr{U}_k$ constructed from D_k by adding one layer of elements from $\mathscr{T}^\eta(\Omega)$ fulfills $D_k \subset\subset \mathscr{U}_k \subset \overline{\Omega_i \cup \Omega_j}$.

Above, $\subset\subset$ means compactly contained. Note that since the regions $\overline{D}_k$ are disjoint and resolved by $\mathscr{T}^\eta(\Omega)$, in Case (c) above, it follows that $\alpha = \alpha_L$ in $\mathscr{U}_k \setminus D_k$. The second condition in (c) avoids that a part of D_k is only "docking". The third condition ensures that D_k passes through the face $\mathscr{F}_{ij}$ only once.

Theorem 5. *Let the above assumptions, in particular Assumption A1, be fulfilled. For the case of classical FETI, assume that for $d = 3$ the intersection of a subdomain with Γ_D is either empty, or contains at least an edge of $\mathscr{T}^\eta(\Omega)$. For the case of TFETI, assume that none of the docking inclusions in Assumption A1(b) intersects the Dirichlet boundary. Then*

$$\kappa_{\mathrm{FETI}} \leq C(\eta)\,(1+\log(\eta/h))^2\,,$$

where $C(\eta)$ is independent of h, the number of subdomains, and α_k, α_L.

The dependence of $C(\eta)$ on η can theoretically be made explicit but is ignored here. In general, it is at least $(H/\eta)^2$. Due to space limitations, we can only give a sketch of the proof for the case of classical FETI; the detailed proof can be found in [12]. To get the condition number bound, we show estimate (6). If $\ker(S_i) = \mathrm{span}\{1\}$, we choose $W_i^\perp := \{w \in W_i : \overline{w}^{\partial\Omega_i} = 0\}$, and $W_i^\perp = W_i$ otherwise. Let $w \in W^\perp$ be arbitrary but fixed. To estimate $|P_D w|_S$, we decompose the interface Γ into *globs* g. These are vertices, edges, or faces of the mesh $\mathscr{T}^\eta(\Omega)$, with one exception: for a face inclusion D_k, we combine all vertices/edges/faces of $\mathscr{T}^\eta(\Omega)$ contained in $\overline{D}_k \cap \Gamma$ into a single glob g. Following [13, Lem. 3.21, Lem. 3.27], we get

$$|(P_D w)_i|_{S_i}^2 \;\leq\; C \sum_{\mathrm{g} \subset \partial\Omega_i \cap \Gamma} \underbrace{\sum_{j \in \mathscr{N}_\mathrm{g} \setminus \{i\}} (\delta_{j|\mathrm{g}}^\dagger)^2\, |I^h(\vartheta_\mathrm{g}(\widetilde{w}_{ii}^\mathrm{g} - \widetilde{w}_{ij}^\mathrm{g}))|_{H^1(\mathsf{U}_{i,\mathrm{g}}),\alpha}^2}_{=:\Upsilon_{i,\mathrm{g}}}\,, \tag{9}$$

where $\vartheta_\mathrm{g} \in V^h(\Omega)$ is a cut-off function (yet to be specified) that equals one on all the nodes on g and vanishes on all other nodes on Γ, I^h is the nodal interpolation operator, and $\mathsf{U}_{i,\mathrm{g}} = \mathrm{supp}(\vartheta_\mathrm{g}) \cap \Omega_i$. The (generic) constant C above only depends the shape regularity constant of $\mathscr{T}^\eta(\Omega)$ and is thus uniformly bounded. For $j \in \mathscr{N}_\mathrm{g}$, the function $\widetilde{w}_{ij}^\mathrm{g} \in V^h(\mathsf{U}_{i,\mathrm{g}})$ is an extension of w_j (yet to be specified) in the sense that $\widetilde{w}_{ij}^\mathrm{g}(x^h) = w_j(x^h)$ for all nodes x^h on g. We treat two cases.

Case 1: g is not part of a face inclusion, i.e., for all $k \in \{1,\dots,n_H\}$ with D_k being a face inclusion, $\overline{D}_k \cap \mathsf{g} = \emptyset$. We choose the cut-off function ϑ_g like in [21, Sect. 4.6] (where the subdomains there are the elements of $\mathscr{T}^\eta(\Omega)$). Using that

$$(\delta^\dagger_{j|\mathsf{g}})^2 \rho_{i|\mathsf{g}} \le \min(\rho_{i|\mathsf{g}}, \rho_{j|\mathsf{g}}) = \alpha_L \qquad \forall j \in \mathscr{N}_\mathsf{g} \setminus \{i\}, \tag{10}$$

and the available techniques from [16, 13], one can show that

$$\Upsilon_{i,\mathsf{g}} \le C \sum_{j\in\mathscr{N}_\mathsf{g}} \alpha_L \left(\omega^2 |\widetilde{w}^\mathsf{g}_{ij}|^2_{H^1(\mathsf{U}_{i,\mathsf{g}})} + \frac{\omega}{\eta^2} \|\widetilde{w}^\mathsf{g}_{ij}\|^2_{L^2(\mathsf{U}_{i,\mathsf{g}})} \right), \tag{11}$$

where above and in the following, $\omega := (1+\log(\eta/h))$.
Case 2: g is part of a face inclusion (see Assumption A1), i.e., there exists k with $\mathsf{g} = \overline{D}_k \cap \Gamma$. Recall that in this case g can be the union of many vertices/edges/faces of $\mathscr{T}^\eta(\Omega)$. We choose a special cut-off function ϑ_g supported in $\mathsf{U}_{i,\mathsf{g}} := \mathscr{U}_k \cap \Omega_i$:

- $\vartheta_\mathsf{g}(x^h) = 1$ for all nodes $x^h \in \overline{D}_k$,
- $\vartheta_\mathsf{g}(x^h) = 0$ for all nodes $x^h \in \partial\mathscr{U}_k \cup (\mathscr{U}_k \cap (\Gamma \setminus \mathsf{g}))$,
- on the elements of the layer, i.e., those elements $T \in \mathscr{T}^\eta(\Omega)$ with $T \subset \mathscr{U}_k \setminus D_k$, we set ϑ_g to the sum of local cut-off functions (similar to Case 1).

By construction, $\vartheta_\mathsf{g} = 1$ on D_k, where $\alpha = \alpha_k$. On the remainder, $\mathscr{U}_k \setminus D_k$, by the assumptions on the coefficient, $\alpha = \alpha_L$. A careful analysis shows that

$$\Upsilon_{i,\mathsf{g}} \le C \sum_{j\in\mathscr{N}_\mathsf{g}} \left(\omega^2 |\widetilde{w}^\mathsf{g}_{ij}|^2_{H^1(\mathsf{U}_{i,\mathsf{g}}),\alpha} + \alpha_L \frac{\omega}{\eta^2} \|\widetilde{w}^\mathsf{g}_{ij}\|^2_{L^2(\Omega_i\cap(\mathscr{U}_k\setminus D_k))} \right). \tag{12}$$

Choice of $\widetilde{w}^\mathsf{g}_{ij}$ in Case 1: We set $\widetilde{w}^\mathsf{g}_{ij} := E^h_{j,\mathsf{g}} \mathscr{H}^{\alpha,h}_j w_j$, where $\mathscr{H}^{\alpha,h}_j : W_j \to V^h(\Omega_j)$ denotes the discrete extension operator such that $|w_j|_{S_j} = |\mathscr{H}^{\alpha,h}_j w_j|_{H^1(\Omega_j),\alpha}$ and $E^h_{j,\mathsf{g}}$ is a suitable transfer operator (see [13, Sect. 2.5.7] or [16, Lem. 5.5]). This results in the estimates

$$|\widetilde{w}^\mathsf{g}_{ij}|_{H^1(\mathsf{U}_{i,\mathsf{g}})} \le C |\mathscr{H}^{\alpha,h}_j w_j|_{H^1(\mathsf{U}'_{j,\mathsf{g}})}, \qquad \|\widetilde{w}^\mathsf{g}_{ij}\|_{L^2(\mathsf{U}_{i,\mathsf{g}})} \le C \|\mathscr{H}^{\alpha,h}_j w_j\|_{L^2(\mathsf{U}'_{j,\mathsf{g}})}. \tag{13}$$

where $\mathsf{U}'_{j,\mathsf{g}} \subset \Omega_j$ is an element of $\mathscr{T}^\eta(\Omega)$ with $\mathsf{g} \subset \overline{\mathsf{U}}'_{j,\mathsf{g}}$.

Choice of $\widetilde{w}^\mathsf{g}_{ij}$ in Case 2: Recall that in this case we are dealing with a face inclusion such that g is part of the face shared by Ω_i and Ω_j and we choose $\mathsf{U}_{i,\mathsf{g}} = \mathscr{U}_k \cap \Omega_j$. To define the extension $\widetilde{w}^\mathsf{g}_{ij} \in V^h(\mathsf{U}_{i,\mathsf{g}})$, we shall combine the technical tools from Sect. 4. Let $\mathsf{U}'_{j,\mathsf{g}} := \mathscr{U}_k \cap \Omega_j$. It can be seen from Assumption A1 that $(\mathsf{U}_{i,\mathsf{g}} \setminus D_k, \mathsf{U}_{i,\mathsf{g}} \cap D_k)$ is a quasi-mirror of $(\mathsf{U}'_{j,\mathsf{g}} \setminus D_k, \mathsf{U}'_{j,\mathsf{g}} \cap D_k)$. We can therefore set

$$\widetilde{w}^\mathsf{g}_{ij} := \Pi^{h,\alpha}_{j,\mathsf{g}} \mathscr{E}^\alpha_{j,\mathsf{g}} \mathscr{H}^{\alpha,h}_j w_j,$$

where $\Pi^{h,\alpha}_{j,\mathsf{g}}$ is the Scott-Zhang interpolator from Lem. 2, $\mathscr{E}^\alpha_{j,\mathsf{g}}$ the extension operator from Lem. 1, and $\mathscr{H}^{\alpha,h}_j$ is defined as above. It has now to be argued that the trans-

formation ϕ in Def. 3 can be chosen such that $\mathscr{E}^{\alpha}_{j,\mathrm{g}}\mathscr{H}^{\alpha,h}_{j}w_j$ is still piecewise linear on the interface $\mathscr{W}'_{j,\mathrm{g}}\cap\partial D_k$. This implies that $\widetilde{w}^{\mathrm{g}}_{ij}$ is indeed an extension of w_j. Due to the properties of the above operators, we obtain the total stability estimates

$$|\widetilde{w}^{\mathrm{g}}_{ij}|_{H^1(\mathsf{U}_{i,\mathrm{g}}),\alpha}\leq C|\mathscr{H}^{\alpha}_{j}w_j|_{H^1(\mathsf{U}'_{j,\mathrm{g}}),\alpha}\,,\quad \|\widetilde{w}^{\mathrm{g}}_{ij}\|_{L^2(\mathsf{U}_{i,\mathrm{g}})}\leq C\|\mathscr{H}^{\alpha}_{j}w_j\|_{L^2(\mathsf{U}'_{j,\mathrm{g}})} \tag{14}$$

for all $w_j\in V^h(\partial\Omega_j)$, with C independent of α_L and α_k. Combining the local estimates (11), (12), (13), and (14), using a finite overlap argument, as well as a *conventional* Poincaré or Friedrichs inequality, one arrives at (6) with $\mu = C\,\omega^2$.

6 Conclusions

Section 3 shows robustness of TFETI for (artificial) coefficients that are quasi-monotone in boundary layers. Sect. 5 shows that these conditions are far from necessary for the robustness of FETI or TFETI. Note that the assumptions and robustness properties of Sect. 5 are similar to the theory in [11] for overlapping Schwarz. Actually, several ideas from the latter theory have been reused in the analysis of Sect. 5. However, the robustness for overlapping Schwarz requires a sophisticated coarse space, whereas for FETI/TFETI, the usual coarse space can be used, which simplifies the implementation a lot.

A combination of the two theories (Sect. 3 and Sect. 5) is of course desirable. However, the general case of α remains open. The problematic cases in FETI/TFETI are certainly (a) a multiple number of inclusions on vertices (or edges in 3D), and (b) long channels that traverse through more than one face, or traverse a face more than once; this is seen in numerical examples; see [12, Sect. 6].

Item (a) might be fixed using suitable FETI-DP/BDDC methods, and we hope that novel analysis of Sect. 5 will have a positive impact here (the known theory of FETI-DP/BDDC for multiscale coefficients is yet limited, cf. [13, 14, 9]). Item (b) can only be addressed by a larger coarse space: either by FETI-DP/BDDC with more sophisticated primal DOFs and/or by spectral techniques as suggested in [20]. Robustness in the spatial scale η is achieved neither in Sect. 3 nor Sect. 5. We believe that the only possibility to gain robustness is a more sophisticated weight selection (cf. Rem. 1) and probably again a larger coarse space.

Acknowledgements The author would like to thank Robert Scheichl, Marcus Sarkis, and Clark Dohrmann for the inspiring collaboration and discussions on this topic.

References

1. Arbogast, T.: Mixed multiscale methods for heterogeneous elliptic problems. In: Numerical Analysis of Multiscale Problems, *LNCSE*, vol. 83, pp. 243–283. Springer, Berlin (2012)

2. Dohrmann, C.R., Pechstein, C.: Constraint and weight selection algorithms for BDDC (2012). Talk at the DD21 Conference, Rennes, June 25; http://www.numa.uni-linz.ac.at/~clemens/dohrmann-pechstein-dd21-talk.pdf
3. Dohrmann, C.R., Widlund, O.B.: Some recent tools and a BDDC algorithm for 3D problems in H(curl). In: R. Bank, R. Kornhuber, O. Widlund (eds.) Domain Decomposition Methods in Science and Engineering XX. LNCSE. Springer, **91**, 15–25 (2013)
4. Dostál, Z., Horák, D., Kučera, R.: Total FETI – An easier implementable variant of the FETI method for numerical solution of elliptic PDE. Commun. Numer. Methods Eng. **22**(12), 1155–1162 (2006)
5. Efendiev, Y., Galvis, J., Lazarov, R., Willems, J.: Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms. ESAIM Math. Model. Numer. Anal. **46**(5), 1175–1199 (2012)
6. Farhat, C., Roux, F.: A method of finite element tearing and interconnecting and its parallel solution algorithm. Int. J. Numer. Meth. Engrg. **32**(6), 1205–1227 (1991)
7. Galvis, J., Efendiev, Y.: Domain decomposition preconditioners for multiscale flows in high contrast media. Multiscale Model. Simul. **8**(4), 1461–1483 (2010)
8. Galvis, J., Efendiev, Y.: Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces. Multiscale Model. Simul. **8**(5), 1621–1644 (2010)
9. Gippert, S., Klawonn, A., Rheinbach, O.: Analysis of FETI-DP and BDDC algorithms for a linear elasticity problems in 3D with compressible and almost incompressible material components. SIAM J. Numer. Anal. **50**(5), 2208–2236 (2012)
10. Graham, I.G., Hagger, M.J.: Unstructured additive Schwarz-conjugate gradient method for elliptic problems with highly discontinuous coefficients. SIAM J. Sci. Comput. **20**(6), 2041–2066 (1999)
11. Graham, I.G., Lechner, P.O., Scheichl, R.: Domain decomposition for multiscale PDEs. Numer. Math. **106**(4), 589–626 (2007)
12. Pechstein, C.: On iterative substructuring methods for multiscale problems. NuMa-Report 2012-13, Institute of Computational Mathematics, Johannes Kepler University (2012). http://www.numa.uni-linz.ac.at/Publications/
13. Pechstein, C.: Finite and Boundary Element Tearing and Interconnecting Solvers for Multiscale Problems, *LNCSE*, vol. 90. Springer-Verlag, Heidelberg (2013)
14. Pechstein, C., Sarkis, M., Scheichl, R.: New theoretical robustness results for FETI-DP. In: R. Bank, R. Kornhuber, O. Widlund (eds.) Domain Decomposition Methods in Science and Engineering XX LNCSE. Springer, **91**, 313–320 (2013)
15. Pechstein, C., Scheichl, R.: Analysis of FETI methods for multiscale PDEs. Numer. Math. **111**(2), 293–333 (2008)
16. Pechstein, C., Scheichl, R.: Analysis of FETI methods for multiscale PDEs. Part II: interface variation. Numer. Math. **118**(3), 485–529 (2011)
17. Pechstein, C., Scheichl, R.: Weighted Poincaré inequalities. IMA J. Numer. Anal. **33**(2), 652–686 (2012)
18. Scheichl, R., Vassilevski, P.S., Zikatanov, L.T.: Mutilevel methods for elliptic problems with highly varying coefficients on non-aligned coarse grids. SIAM J. Numer. Anal. **50**(3), 1675–1694 (2012)
19. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R.: Abstract robust coarse spaces for systems of PDEs via gereralized eigenproblems in the overlap. Numer. Math. (2013). Accepted; preprint: NuMa-Report 2011-07, JKU Linz
20. Spillane, N., Rixen, D.: Automatic spectral coarse spaces for robust FETI and BDD algorithms. Int. J. Numer. Meth. Engrg. (2013). Accepted; http://hal.archives-ouvertes.fr/hal-00756994
21. Toselli, A., Widlund, O.B.: Domain Decomposition Methods – Algorithms and Theory. Springer, Berlin (2005)
22. Xu, J., Zhu, Y.: Uniform convergent multigrid methods for elliptic problems with strongly discontinuous coefficients. Math. Mod. Meth. Appl. Sci. **18**(1), 77–105 (2008)

A Mortar BDD method for solving flow in stochastic discrete fracture networks

Géraldine Pichot[1], Baptiste Poirriez[2], Jocelyne Erhel[1], and Jean-Raynald de Dreuzy[3]

1 Introduction

In geological media, the large variety and complex configurations of fractured networks make it difficult to describe them precisely. A relevant approach is to model them as Discrete Fracture Networks (DFN)[10, 19], with statistical properties in agreement with in situ experiments [15, 13, 14]. A DFN is a 3D domain made of 2D fractures intersecting each other. Steady state flow in DFN is considered, the rock matrix is assumed impervious. Following a Monte-Carlo approach, a large number of DFN has to be generated and for each, a flow problem has to be solved whatever the complexity of the generated networks. Moreover time and memory costs for each simulation should be as lower as possible.

A nonconforming discretization of DFN allows to reduce the number of unknowns and facilitate mesh refinement. Sharp angles are managed by a staircase-like discretizations of the fractures' contours [34]. The non-matching feature at the fractures' intersections is handled via a Mortar method [4, 5, 1] developed for DFN in [33, 34] for a mixed hybrid finite element formulation. It consists in defining, for each intersection between fractures, master and slave sides. Due to the staircase-like discretizations, a shared edge may be labeled several times with master and/or slave properties, it is called in the paper a multi-labeled edge. Continuity conditions are enforced between the unknowns on both sides. The derived linear system has only inner and master traces of hydraulic head as unknowns. The matrix A of this system is a symmetric definite positive (SPD) arrow matrix in presence of Dirichlet boundary conditions [34].

The challenge is to solve such linear systems with millions of unknowns [17]. Direct solvers (like Cholmod [11]) are very efficient for small systems but suffer from a high need of RAM memory when the system size becomes too large. Among iterative solvers, multigrid methods are very efficient for most networks but for some, the convergence rate is very slow [35, 17]. Preconditioned Conjugate Gradient (PCG) is efficient and robust for every network tested [35]. The natural decomposition of the matrix A in subdomains encourages the use of domain decomposition methods [7, 36, 31, 24]. The Schur complement of the matrix A is SPD and yields an interface

[1] INRIA Rennes Bretagne-Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, e-mail: {geraldine.pichot}{jocelyne.erhel}@inria.fr .[2] IRISA, Campus de Beaulieu, 35042 Rennes Cedex, e-mail: baptiste.poirriez@irisa.fr .[3] Géosciences Rennes, Campus de Beaulieu, 35042 Rennes Cedex, e-mail: Jean-Raynald.de-Dreuzy@univ-rennes1.fr

system with only master unknowns. This interface system can be solved iteratively with PCG. The unknowns on inner edges are then derived locally in each fracture plane by solving small local linear systems, with a direct solver for example.

Among possible preconditioners, the balancing domain decomposition (BDD) method is based on a Neumann-Neumann preconditioner coupled with a coarse level solver, to improve the preconditioner as the number of subdomains increases [29, 30, 27]. BDD method applied to mixed finite element is done in [12]. The application to a nonconforming discretization is proposed in [18, 32]. Meanwhile, an alternative method has been developed, the Balancing Domain Decomposition by Constraints (BDDC) [16], later applied to mortar discretization for geometrically nonconforming partitions in [26].

In this paper, we use the BDD algorithm proposed in [32, 35] to solve the linear system arising from a nonconforming discretization of DFN. The coarse level is defined following [37] and balancing is implemented as a preconditioning matrix [21]. The algorithm is implemented in C++ in the parallel software SIDNUR [35]. For DFN, choosing one subdomain given by one fracture, instead of a set of fractures has shown to be the most time saving decomposition [35].

The paper is organized in four sections. Section 2 describes the flow model. Section 3 recalls the linear system derived from a nonconforming discretization of the DFN. Section 4 is the main contribution of this paper and presents the decomposition in local matrices. We apply the BDD method proposed in [32, 35] for networks satisfying some hypotheses on the mesh. The last section illustrates the application of the solver SIDNUR [35] on three stochastically generated DFN.

2 Flow model

We consider flow in DFN assuming the rock matrix is impervious. In the entire paper, an intersection is uniquely defined as the segment shared by two fractures. We denote Σ_k the kth intersection, $k = 1, ..., N_i$.

Poiseuille's law and mass conservation apply in each fracture plane, denoted Ω_f, $f = 1, ..., N_f$. We assume there is no longitudinal flux at the fracture intersections.

The DFN is embedded in a cube of size L. Some fractures are truncated by the cube faces. Classical permeameter boundary conditions apply on the cube faces. The two opposite faces of the cube with Dirichlet boundary conditions (prescribed value p^D) are called Γ_D ($\Gamma_D \neq \emptyset$) and the lateral faces with homogeneous Neumann boundary conditions are called Γ_N. The boundary of the fracture f is called Γ_f. In the following, we assume there is only one cluster of fractures connected to the Dirichlet boundary conditions and we consider only this cluster.

In each fracture plane, with $x \in \mathbb{R}^2$, the following equations link the unknown hydraulic head scalar function $p(x)$ and the flux per unit length function $u(x)$:

$$\nabla \cdot u(x) = f(x) \quad \text{for } x \in \Omega_f, \tag{1}$$

$$u(x) = -\mathscr{T}(x)\nabla p(x) \quad \text{for } x \in \Omega_f, \tag{2}$$

$$p(x) = p^D(x) \quad \text{on } \Gamma_D \cap \Gamma_f, \tag{3}$$
$$u(x).\nu = 0 \quad \text{on } \Gamma_N \cap \Gamma_f, \tag{4}$$
$$u(x).\mu = 0 \quad \text{on } \Gamma_f \backslash \{(\Gamma_f \cap \Gamma_D) \cup (\Gamma_f \cap \Gamma_N)\}, \tag{5}$$

where ν (respectively μ) denotes the outward normal unit vector of the borders with respect to the fracture Ω_f. The parameter $\mathscr{T}(x)$ is a given SPD transmissivity field (unit $[\mathrm{m}^2.\mathrm{s}^{-1}]$). The function $f(x) \in L^2(\Omega_f)$ represents the sources/sinks.

Let $\mathscr{I}_l$ be a segment shared by several incident fractures, $l = 1, ..., N_l$. It can be the intersection itself or only a part of it if intersections overlap. Let F_l be the set of fractures which contains $\mathscr{I}_l$. On each segment, continuity conditions are imposed to ensure the continuity of hydraulic heads and the conservation of fluxes [20], [38]:

$$p_{f,l} = p_l \quad \text{on } \mathscr{I}_l, \forall f \in F_l, \tag{6}$$
$$\sum_{f \in F_l} u_{f,l}.n_{f,l} = 0 \quad \text{on } \mathscr{I}_l, \tag{7}$$

where $p_{f,l}$ is the trace of hydraulic head on $\mathscr{I}_l$ in the fracture Ω_f, p_k is the unknown hydraulic head on the segment $\mathscr{I}_l$ and $u_{f,l}.n_{f,l}$ is the normal flux through $\mathscr{I}_l$ coming from the fracture Ω_f, with $n_{f,l}$ the outward normal unit vector of the segment $\mathscr{I}_l$ with respect to the fracture Ω_f.

3 A Mortar method applied to DFN

3.1 Mesh generation

With a stochastic generation, fractures can cross in a very intricate way. We define the contour of a fracture f as its border and all segments $\mathscr{I}_l$ which belong to f. To preserve a good mesh quality whatever the generated fractured networks, staircase like discretizations of the contour are performed in each fracture plane.

Each fracture is meshed with its own mesh step:

(i) A temporary uniform grid is built that encompasses the fracture, with a grid step chosen as input;
(ii) 1D staircase-like meshes of the contour are built using the centers of the grid elements as discretization points;
(iii) From these 1D discretizations, a 2D triangle mesh of the fracture is built.

We call shared edges the edges of the triangles that discretize the segments $\mathscr{I}_l$, $l = 1, ..., N_l$ within the different fractures in F_l. All other edges are called inner edges. Notice a given segment $\mathscr{I}_l$ may have different discretizations in the different fractures in F_l as shown on figure 1. The total mesh is made of N_{in} inner edges and of N_Σ shared edges. In the following, we will use the subscript *in* to refer to the inner edges and Σ to shared edges.

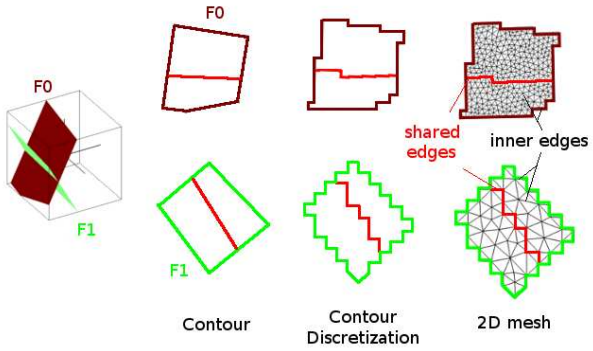


Fig. 1 Mesh generation - Simple example with two fractures

3.2 Derivation of the linear system

The Mortar method applied to DFN is presented in [34]. It consists, for each intersection Σ_k, of choosing a master fracture m and a slave fracture s. We denote $N_m = \sum_{k=1}^{N_i} N_{k,m}$, $N_s = \sum_{k=1}^{N_i} N_{k,s}$, with $N_{k,\{m,s\}}$ the number of edges that discretize the master (respectively slave) side of the intersection Σ_k.

The traces of hydraulic head unknowns are Λ_{in} on inner edges, Λ_m and Λ_s on master and slave edges. Additionnally, each shared edge has an unknown called Λ_Σ. The additional unknowns Λ_Σ allow to deal with multi-labeled edges which belong to several intersections. The unknowns Λ_s and Λ_Σ are derived from Λ_m following the relations (see [34]):

$$\Lambda_s = C\Lambda_m, \tag{8}$$

$$\Lambda_\Sigma = P_m\Lambda_m + P_s\Lambda_s = (P_m + P_sC)\Lambda_m. \tag{9}$$

The matrix C is an intersection block matrix of dimension N_sxN_m, with the block C_k a matrix of size $N_{k,s}$x$N_{k,m}$ for the intersection Σ_k that represents the L^2-projection from the master side to the slave side.

Let denote m_E (respectively s_E) the number of times a shared edge E is labeled with a master (respectively slave) property. Let $n_E = s_E + m_E$. The values (i, j) of the matrices P_m (respectively P_s) of size N_ΣxN_m (respectively N_ΣxN_s) is $\frac{1}{n_E}$ if the unknown $\Lambda_m(j)$ (respectively $\Lambda_s(j)$) is associated to an edge with $\Lambda_\Sigma(i)$ as shared unknown, and 0 otherwise.

At the network scale, the linear system reduces to a system with unknowns Λ_{in} and Λ_m [34]:

$$A\begin{pmatrix}\Lambda_{in}\\ \Lambda_m\end{pmatrix}=\begin{pmatrix}F_{in}\\ F_m\end{pmatrix}. \tag{10}$$

The second member is a vector of dimension $N_{in}+N_m$, which corresponds to the source/sink function, to the imposed Dirichlet and Neumann boundary conditions.

The matrix A is SPD in presence of Dirichlet boundary conditions [34] and writes as:

$$\begin{cases} A &= \begin{pmatrix} A_{in,in} & A_{in,m}\\ A_{in,m}^T & A_{m,m}\end{pmatrix},\\ A_{in,m} &= A_{in,\Sigma}(P_m+P_sC),\\ A_{m,m} &= (P_m+P_sC)^T A_{\Sigma,\Sigma}(P_m+P_sC).\end{cases} \tag{11}$$

The matrix $A_{in,in}$ is a block diagonal matrix of order N_{in} made of blocks $A_{f,in,in}$ associated to the inner edges in the fracture Ω_f.

4 A Mortar BDD method for DFN system

The arrow shape of the matrix A allows to reduce the linear system (10) to an interface problem with only Λ_m as unknowns:

$$S\Lambda_m = B_m, \tag{12}$$

$$S = A_{m,m} - A_{in,m}^T A_{in,in}^{-1} A_{in,m}, \tag{13}$$

$$B_m = F_m - (P_m^T + C^T P_s^T) A_{in,\Sigma}^T A_{in,in}^{-1} F_{in}. \tag{14}$$

with S the Schur complement of size N_mxN_m.

Since S is SPD, the linear system (12) can be solved iteratively via a PCG method. To apply a balancing preconditioner, we need the local Schur complements S_f, $f=1,...,N_f$.

4.1 Local Schur complements

Let $N_{f,m}$ (respectively $N_{f,s}$) be the number of master (respectively slave) unknowns associated with master (respectively slave) edges in the fracture f. Let $N_{f,o}$ be the number of master unknowns associated with the slave edges in the fracture f following the relations (8). Let $N_{f,\Sigma}$ be the number of shared edges in the fracture f. We define the local matrices $(P_m+P_sC)_f$ as:

$$(P_m+P_sC)_f = \begin{pmatrix} P_{f,m} & P_{f,s}C_f \end{pmatrix} \tag{15}$$

with $P_{f,m}$ of size $N_{f,\Sigma}$x$N_{f,m}$ and $P_{f,s}$ of size $N_{f,\Sigma}$x$N_{f,s}$. The matrix C_f of size $N_{f,s}$x$N_{f,o}$ is a block matrix whose blocks C_k are extracted from the matrix C for the intersections Σ_k in the fracture f.

The local problem in the fracture f writes as:

$$A_{f,\Sigma} = \begin{pmatrix} A_{f,in,in} & A_{f,in,\Sigma} \\ A_{f,in,\Sigma}^T & A_{f,\Sigma,\Sigma} \end{pmatrix} \tag{16}$$

Its associated Schur complement writes as: $S_{f,\Sigma} = A_{f,\Sigma,\Sigma} - A_{f,in,\Sigma}^T A_{f,in,in}^{-1} A_{f,in,\Sigma}$.

At the fracture scale, local matrices A_f, of order $(N_{f,in} + N_{f,m} + N_{f,o})$ are built from $A_{f,\Sigma}$:

$$\begin{cases} A_f &= \begin{pmatrix} A_{f,in,in} & A_{f,in,m} \\ A_{f,in,m}^T & A_{f,m,m} \end{pmatrix}, \\ A_{f,in,m} &= \begin{pmatrix} A_{f,in,\Sigma} P_{f,m} & A_{f,in,\Sigma} P_{f,s} C_f \end{pmatrix}, \\ A_{f,m,m} &= \begin{pmatrix} P_{f,m}^T A_{\Sigma,\Sigma} P_{f,m} & P_{f,m}^T A_{\Sigma,\Sigma} P_{f,s} C_f \\ (P_{f,m}^T A_{\Sigma,\Sigma} P_{f,s} C_f)^T & (P_{f,s} C_f)^T A_{\Sigma,\Sigma} P_{f,s} C_f \end{pmatrix}. \end{cases} \tag{17}$$

The block $A_{f,in,m}$ is of size $N_{f,in}$x$(N_{f,m} + N_{f,o})$ and the block $A_{f,m,m}$ is of size $(N_{f,m} + N_{f,o})$x$(N_{f,m} + N_{f,o})$.

The local Schur complement S_f associated to the matrix A_f (17) of the fracture Ω_f writes:

$$S_f = A_{f,mm} - A_{f,in,m}^T A_{f,in,in}^{-1} A_{f,in,m} = (P_m + P_s C)_f^T S_{f,\Sigma} (P_m + P_s C)_f. \tag{18}$$

As each intersection involves two fractures, one slave and one master, the Schur complement S of size N_mxN_m is the sum of the local Schur complements:

$$S = \sum_{f=1}^{N_f} R_f^T S_f R_f, \tag{19}$$

where R_f is the rectriction matrix from the network to the fracture f.

4.2 Neumann-Neumann preconditioner

In the following, a subdomain Ω_f is said to be floating if it does not contain any Dirichlet boundary conditions, non floating otherwise.

The Neumann-Neumann preconditioner [25, 9, 28] writes as:

$$M_{NN}^{-1} = D \sum_f R_f^T S_f^\dagger R_f D, \tag{20}$$

where

$$S_f^\dagger = \begin{cases} S_f^{-1} \text{ if } S_f \text{ is non singular,} \\ \tilde{S}_f^{-1} \text{ otherwise, with } \tilde{S}_f \text{ a non singular approximation of } S_f. \end{cases} \tag{21}$$

The matrix D is a diagonal matrix of order N_m. With a nonconforming discretization, a definition of one fracture as one subdomain and an homogeneous transmissivity, $D = \frac{1}{2}Id$ since each master unknown is defined for an intersection between two subdomains.

From the definition of M_{NN}^{-1}, one needs to solve local subdomain problems with the matrix S_f, like $S_f z_f = r_f$. However the kernel of S_f may not be trivial. If the matrix $(P_m + P_s C)_f$ is of full rank, the kernel of S_f is that of $S_{f,\Sigma}$: $\{0\}$ for a non floating subdomain, else $\{const\}$. We assume that $(P_m + P_s C)_f$ is of full rank if the following conditions are satisfied:

(H1) the master side of an intersection must have the smallest number of discretization edges: $N_{k,m} \leq N_{k,s}, \forall k \in 1, ..., N_i$;

(H2) There are no multi-labeled edges: $n_E = 1$ for each shared edge E yielding: $N_\Sigma = N_m + N_s$.

If the subdomain is floating, in order to get a SPD approximation $\tilde{S}_f$, we add one arbitrary Dirichlet condition, since the kernel is of dimension 1 [35].

4.3 Balancing preconditioner

As the number of subdomains increases, the efficiency of the Neumann-Neumann preconditioner decreases [27] and one has to couple it with a coarse level solver [29, 30]. We use the following balancing preconditioner:

$$M_b^{-1} = P^T M_{NN}^{-1}, \tag{22}$$

as in [37, 21, 35] where the projection matrix P, of order N_m, is defined as:

$$P = I - SZS_c^{-1}Z^T. \tag{23}$$

The matrix Z is a N_mx N_c subspace matrix with full rank, $N_c < N_m$, and $S_c = Z^T SZ$ is the invertible matrix corresponding to the coarse problem.

This formulation is based on the PCG initial value:

$$\Lambda_{m,0} = ZS_c^{-1}Z^T B_m, \tag{24}$$

such that, for all iterations *it* of PCG, the residuals $r_{it} = S\Lambda_{m,it} - B_m$ satisfy $Z^T r_{it} = 0$ and $Pr_{it} = r_{it}$ [35]. Thus applying (22) is equivalent to apply $P^T M_{NN}^{-1} P + ZS_c^{-1}Z^T$ [37, 35].

A possible choice for the full rank matrix Z is to use a subdomain deflation as defined in [22, 35]. Here $N_c \leq N_f$ and Z is sparse.

5 Numerical experiments

We present preliminary numerical experiments on three random DFN that satisfy hypotheses $(H1)-(H2)$, generated with the software MP_FRAC of the H2OLab platform `http://h2olab.inria.fr/`. We checked there is only one connected cluster. We build the local matrices A_f and use the software SIDNUR which implements the BDD method [35].

5.1 Geometry and boundary conditions

The position of the fractures is taken as uniform in the domain. Their orientation is uniform and their length follows a power law distribution of exponent 2.7 [8]. We take $p^D = 1m$ on the cube face at $y = L/2$ and $p^D = 20m$ on the cube face at $y = -L/2$. The transmissivity tensor is homogeneous and equal to $\mathscr{T} = T\,Id$, with $T = 8.2e-7$ m^2.s^{-1}. We consider 3 networks:

- $L6\_NF28$: L=6 and N_f=28;
- $L10\_NF18$: L=10 and N_f=18;
- $L10\_NF24$: L=10 and N_f=24.

5.2 Mesh procedure and basic optimization

The nonconforming mesh is generated according to the mesh procedure described in subsection 3.1. With this approach, adaptative mesh refinement can be done at the fracture level [2, 3, 39, 6].

A basic mesh coarsening consists in meshing finely only the fractures that take part significantly in the flow. Let us run a first simulation with a coarse mesh step $2*\Delta$. The output flux for each fracture is computed, as well as the total output flux on the output cubic face. We choose to refine, with a mesh step Δ, the fractures that have an output flux above 5 % of the total output flux. The simulation is performed again on this refined mesh.

In table 1, we compare the mesh obtained with this basic mesh coarsening, so-called coarser mesh, with a mesh where the step is Δ for all fractures, so-called fine mesh. The min and mean of the quality mesh criterion $Q_K \in [0;1]$ is also given, where Q_K is defined for each triangle K as [23]:

$$Q_K = 4\sqrt{3}\frac{S_K}{h_s^2}, \tag{25}$$

with S_K the surface of the triangle K and $h_s = \sqrt{\sum_{i=1}^{3} h_i^2}$, with h_i the length of the edge i of the triangle K. The closer Q_K is to 1, the better the triangle quality is.

Table 1 Comparison between a mesh with step Δ for all fractures and a mesh with step Δ for fractures with an output flux above 5 % of the total output flux and $2*\Delta$ otherwise

Simulation name	Δ	Fine mesh - step Δ			Coarser mesh - step Δ or 2Δ		
		Number of edges	Min(Q_K)	Mean(Q_K)	Number of edges	Min(Q_K)	Mean(Q_K)
$L6\_NF28$	0.05	122306	0.43	0.95	90533	0.23	0.95
$L10\_NF18$	0.1	62409	0.45	0.95	57462	0.19	0.95
$L10\_NF24$	0.1	78652	0.51	0.95	67765	0.25	0.95

Table 1 shows that this basic mesh coarsening reduces the number of edges from 7.93 % to 25.96 % at the price of somehow lower mesh quality. Indeed the length of some fractures is too small compared with 2Δ, yielding too few discretization points. As future work, we could define a minimal mesh step per fracture according to its length.

5.3 Solution with SIDNUR

Using the coarser mesh, we solve the linear system (12) with the BDD method. We checked these networks satisfy hypotheses $(H1)-(H2)$. From the computed values of Λ_m, we derive the unknowns Λ_s and Λ_Σ according to $(8)-(9)$. The inner unknowns Λ_{in} are derived locally in each fracture plane by solving small linear systems (see (10)). From these traces of hydraulic head unknowns, one can derive the mean head values and the fluxes [34]. Figures 2, 3 and 4 give the mean head values on the three DFN. Figure 5 displays the mean head values for the DFN $L10\_NF24$ obtained by solving the linear system (12) with CHOLMOD to illustrate the good agreement of the results obtained with the two methods.

Table 2 gives the numbers N_{in}, N_m and N_s with $N_\Sigma = N_m + N_s$ (hypothesis $(H2)$). This table also provides the number of PCG iterations, the final L^2-norm of the residual and the L^2-norm of the relative difference between the solutions $\begin{pmatrix}\Lambda_{in}\\ \Lambda_m\end{pmatrix}$ computed with SIDNUR and with the direct solver CHOLMOD [11].

Table 2 Solution with SIDNUR. Comparison with CHOLMOD

Simulation name	N_{in}	N_m	N_s	# PCG it.	PCG final residual	Comparison with CHOLMOD
$L6\_NF28$	89732	365	436	13	6.02e-17	4.15e-12
$L10\_NF18$	56939	247	276	15	2.47e-18	9.56e-13
$L10\_NF24$	66899	412	454	18	8.71e-19	1.47e-12

On such small linear systems with very small CPU times, the solver SIDNUR is not competitive with respect to a direct solver. However this preliminary test phase demonstrates the possibility of solving linear system arising from a nonconforming discretization of networks satisfying hypotheses $(H1)-(H2)$ with the BDD

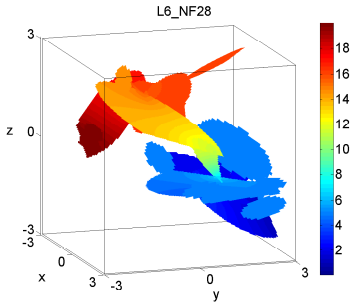


Fig. 2 $L6\_NF28$ - Mean head - SIDNUR

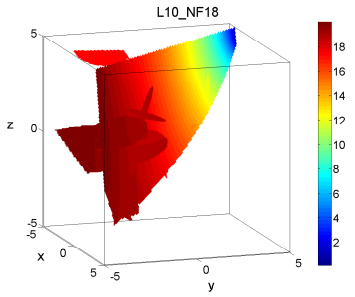


Fig. 3 $L10\_NF18$ - Mean head - SIDNUR

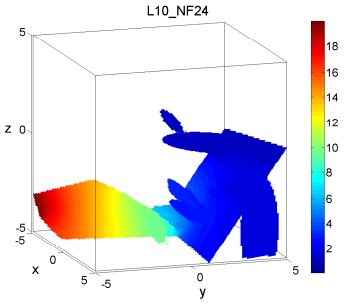


Fig. 4 $L10\_NF24$ - Mean head - SIDNUR

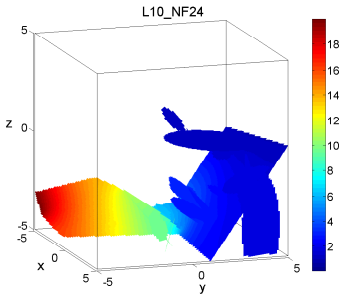


Fig. 5 $L10\_NF24$ - Mean head - CHOLMOD

method. Using SIDNUR relies on a suitable decomposition of the local matrices. Moreover SIDNUR requires less RAM memory than a direct solver and is parallel.

6 Conclusion

This paper describes a Balancing Domain Decomposition method, implemented in the so-called SIDNUR solver, to simulate flow in DFN with a nonconforming mesh. DFN and local matrices are generated with the so-called MP_FRAC software. Our current work is to extend the method to more general discretizations, which do not satisfy hypotheses $(H1) - (H2)$, in the perspective of solving linear systems with several millions of unknowns. The parallelism of SIDNUR will be very helpful to reduce the time and memory costs. Moreover the very basic technic we use to coarsen the mesh could be improved by defining suitable *a posteriori* estimators.

Acknowledgements This work was supported by the French National Research Agency, with the ANR-07-CIS7 project MICAS, and by INRIA with the ARC-INRIA GEOFRAC project.

References

1. Arbogast, T., Cowsar, L.C., Wheeler, M.F., Yotov, I.: Mixed finite element methods on nonmatching multiblock grids. SIAM J. Numer. Anal **37**, 1295–1315 (2000)
2. Arnold, D.N., Brezzi, F.: Mixed and nonconforming finite element methods : implementation, postprocessing and error estimates. ESAIM: Mathematical Modelling and Numerical Analysis - Modlisation Mathmatique et Analyse Numrique **19**(1), 7–32 (1985). URL `http://eudml.org/doc/193443`
3. Bernardi, C., Hecht, F.: Error indicators for the mortar finite element discretization of the Laplace equation. Mathematics of Computation **71**(240), 1371–1403 (2002)
4. Bernardi, C., Maday, Y., Patera, A.T.: Domain decomposition by the mortar element method. In: Asymptotic and numerical methods for partial differential equations with critical parameters (Beaune, 1992), *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, vol. 384, pp. 269–286. Kluwer Acad. Publ., Dordrecht (1993)
5. Bernardi, C., Maday, Y., Patera, A.T.: A new nonconforming approach to domain decomposition: the mortar element method. In: Nonlinear partial differential equations and their applications. Collège de France Seminar, Vol. XI (Paris, 1989–1991), *Pitman Res. Notes Math. Ser.*, vol. 299, pp. 13–51. Longman Sci. Tech., Harlow (1994)
6. Bernardi, C., Rebollo, T.C., Hecht, F., Mghazli, Z.: Mortar finite element discretization of a model coupling darcy and stokes equations. ESAIM: Mathematical Modelling and Numerical Analysis **42**(3), 375–410 (2008). URL `http://eudml.org/doc/250402`
7. Bjørstad, P.E., Widlund, O.B.: Solving elliptic problems on regions partitioned into substructures. In: G. Birkhoff, A. Schoenstadt (eds.) Elliptic Problem Solvers II, pp. 245–256. Academic Press, New York (1984)
8. Bonnet, E., Bour, O., Odling, N., Davy, P., Main, I., Cowie, P., Berkowitz, B.: Scaling of fracture systems in geological media. Reviews of Geophysics **39**(3), 347–383 (2001)
9. Bourgat, J.F., Glowinski, R., Le Tallec, P., Vidrascu, M.: Variational formulation and algorithm for trace operator in domain decomposition calculations. In: T. Chan, R. Glowinski, J. Périaux, O. Widlund (eds.) Domain Decomposition Methods, pp. 3–16. SIAM, Philadelphia, PA (1989)
10. Cacas, M.C., Ledoux, E., de Marsily, G., Barbeau, A., Calmels, P., Gaillard, B., Magritta, R.: Modeling fracture flow with a stochastic discrete fracture network: calibration and validation. 1. the flow model. Water Resources Research **26**(3), 479–489 (1990)
11. Chen, Y., Davis, T.A., Hager, W.W., Rajamanickam, S.: Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. ACM Trans. Math. Softw. **35**(3), 22:1–22:14 (2008). DOI 10.1145/1391989.1391995. URL `http://doi.acm.org/10.1145/1391989.1391995`
12. Cowsar, L.C., Mandel, J., Wheeler, M.F.: Balancing domain decomposition for mixed finite elements. Mathematics of computation **64**(211), 989–1015 (1995). DOI {10.2307/2153480}
13. Davy, P., Bour, O., de Dreuzy, J.R., Darcel, C.: Flow in multiscale fractal fracture networks. In: 261 (ed.) Geological society, London, special publications, pp. 31–45 (2006)
14. Davy, P., Le Goc, R., Darcel, C., Bour, O., de Dreuzy, J.R., Munier, R.: A likely universal model of fracture scaling and its consequence for crustal hydromechanics. Journal of Geophysical Research **115**, B10,411 (2010)
15. Dershowitz, W.S., Einstein, H.H.: Characterizing rock joint geometry with joint system models. Rock Mechanics and Rock Engineering **21(1)**, 2151 (1988)
16. Dohrmann, C.R.: A preconditioner for substructuring based on constrained energy minimization. SIAM J. Sci. Comput. **25**, 246–258 (2003)
17. de Dreuzy, J.R., Pichot, G., Poirriez, B., Erhel, J.: Synthetic benchmark for modeling flow in 3D fractured media. Computers & Geosciences **50**, 59–71 (2013). DOI 10.1016/j.cageo.2012.07.025. URL `http://hal.inria.fr/hal-00735675`
18. Dryja, M., Proskurowski, W.: On preconditioners for mortar discretization of elliptic problems. Numerical Linear Algebra with Applications **10**(1-2), 65–82 (2003). DOI 10.1002/nla.312. URL `http://dx.doi.org/10.1002/nla.312`

19. Dverstop, B., Andersson, J.: Application of the discrete fracture network concept with field data: possibilities of model calibration and validation. Water Resources Research **25(3)**, 540550 (1989)
20. Erhel, J., de Dreuzy, J.R., Poirriez, B.: Flow simulations in three-dimensional discrete fracture networks. SIAM Journal on Scientific Computing **31**(4), 2688–2705 (2009). DOI DOI:10.1137/080729244
21. Erhel, J., Guyomarc'h, F.: An augmented conjugate gradient method for solving consecutive symmetric positive definite systems. SIAM Journal on Matrix Analysis and Applications **21**(4), 1279–1299 (2000)
22. Frank, J., Vuik, C.: On the construction of deflation-based preconditioners. SIAM J. Sci. Comput. **23**(2), 442–462 (2001). DOI 10.1137/S1064827500373231. URL http://dx.doi.org/10.1137/S1064827500373231
23. Frey, P.J., George, P.L.: Maillages : applications aux éléments finis. Hermès sciences publ. DL1999 (53-Mayenne), Paris. URL http://opac.inria.fr/record=b1094298
24. Gander, M.J., Tu, X.: On the origins of iterative substructuring methods. In: Domain Decomposition Methods (2013)
25. Glowinski, R., Wheeler, M.F.: Domain decomposition and mixed finite element methods for elliptic problems. In: R. Glowinski, G.H. Golub, G.A. Meurant, J.P. Eds (eds.) Domain Decomposition Methods for Partial Differential Equations, pp. 144–172. SIAM, Philadelphia (1988)
26. Kim, H.H., Dryja, M., Widlund, O.B.: A bddc method for mortar discretizations using a transformation of basis. SIAM J. Numerical Analysis **47**(1), 136–157 (2008)
27. Le Tallec, P.: Domain decomposition methods in computational mechanics. In: J.T. Oden (ed.) Computational Mechanics Advances, vol. 1 (2), pp. 121–220. North-Holland (1994)
28. Le Tallec, P., De Roeck, Y., Vidrascu, M.: Domain decomposition methods for large linearly elliptic three-dimensional problems. Journal of Computational and Applied Mathematics **34**, 93–117 (1991)
29. Mandel, J.: Balancing domain decomposition. Communications in Applied Numerical Methods **9**, 233241 (1993)
30. Mandel, J., Brezina, M.: Balancing domain decomposition: Theory and computations in two and three dimensions. Tech. Rep. UCD/CCM 2, Center for Computational Mathematics, University of Colorado at Denver (1993)
31. Meurant, G.: Computer Solution of Large Linear Systems. Elsevier Science B.V. (1999)
32. Pencheva, G., Yotov, I.: Balancing domain decomposition for mortar mixed finite element methods. Numerical Linear Algebra with Applications **10**(1-2), 159–180 (2003). DOI 10.1002/nla.316. URL http://dx.doi.org/10.1002/nla.316
33. Pichot, G., Erhel, J., de Dreuzy, J.R.: A mixed hybrid mortar method for solving flow in discrete fracture networks. Applicable Analysis **89**(10), 1629–1643 (2010)
34. Pichot, G., Erhel, J., de Dreuzy, J.R.: A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks. SIAM Journal on Scientific Computing **34(1)**, B86B105. (20 pages) (2012)
35. Poirriez, B.: Etude et mise en oeuvre d'une méthode de sous-domaines pour la modélisation de l'écoulement dans des réseaux de fractures en 3d. Ph.D. thesis, University of Rennes 1 (2011)
36. Quarteroni, A., Valli, A.: Domain Decomposition Methods for Partial Differential Equations. Oxford Science Publications (1999)
37. Tang, J.M., Nabben, R., Vuik, C., Erlangga, Y.A.: Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. J. Sci. Comput. **39**(3), 340–370 (2009). DOI 10.1007/s10915-009-9272-6. URL http://dx.doi.org/10.1007/s10915-009-9272-6
38. Vohralik, M., Maryska, J., Severyn, O.: Mixed and nonconforming finite element methods on a system of polygons. Applied Numerical Mathematics **57**, 176–193 (2007)
39. Wheeler, M.F., Yotov, I.: A posteriori error estimates for the mortar mixed finite element method. SIAM J. Numer. Anal. **43**(3), 1021–1042 (2005). DOI 10.1137/S0036142903431687. URL http://dx.doi.org/10.1137/S0036142903431687

A Domain-Based Multinumeric Method for the Steady-State Convection-Diffusion Equation

Beatrice Riviere[1] and Xin Yang[1]

1 Introduction

In the simulation of flow and transport of hydrocarbons in reservoirs, locally mass conservative methods are preferred. Methods that do not satisfy this property, will produce numerical mass errors that accumulate and will yield an unstable solution. Currently, finite volume methods are popular numerical methods in the oil industry. While they are computationally efficient, they are only of first order. Convergence of cell-centered finite volume solutions is theoretically obtained on specially constructed grids (such as Voronoi meshes) and for problems with no mixed second derivatives [3, 4, 8, 12, 6]. Discontinuous Galerkin methods also belong to the class of locally mass conservative methods. In addition, their flexibility allows for the use of complicated geometries, unstructured meshes, varying polynomial degrees and discontinuous coefficients. Discontinuous Galerkin solutions are accurate but their cost can be large as it is proportional to the the number of mesh elements (also called cells). In this paper, discontinuous Galerkin methods are used in certain parts of the domain whereas the cell-centered finite volume method is used in other parts. The model problem is a convection-diffusion problem in a bounded domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$.

$$-\nabla \cdot (K\nabla u - \beta u) = f, \quad \text{in} \quad \Omega, \tag{1}$$

$$u = g, \quad \text{on} \quad \partial\Omega. \tag{2}$$

The spatially dependent coefficient K is bounded below and above by positive constants k_0 and k_1 respectively. The convective vector β is assumed to be divergence-free: $\nabla \cdot \beta = 0$.

The computational domain is partitioned into several subdomains. On each subdomain, either a discontinuous Galerkin method is used or a cell-centered finite volume is used. The advantage of a multinumeric approach lies in the ability of choosing a particular scheme for a particular subdomain. The discontinuous Galerkin method can yield accurate solutions in parts of the domain where the permeability of the porous medium varies over several orders of magnitude or in parts of the domain where anisotropy is important. In this work, the coupling of the two discretizations is done weakly by interface conditions. Two equivalent formulations are presented: a monolithic approach and an hybridized approach with Lagrange multipliers. This paper extends the result of [2] where the elliptic problem is analyzed. In [11], we apply the method to a transport equation. The idea of using different discretizations

[1] Rice University, 6100 Main Street, Houston, Texas 77005, USA, e-mail: riviere@caam.rice.edu e-mail: xin.yang@rice.edu.

in different subdomains is well studied in the literature. For instance, the reader can refer to [1, 5, 10, 7].

An outline of the paper is the following. Section 2 defines first the discontinuous Galerkin and finite volume discretizations in each subdomain, then the coupling of the subdomains. Section 3 states the convergence of the method. Conclusions follow.

2 A multinumeric approach

The domain Ω is subdivided into non-overlapping subdomains Ω^i_{FV} and Ω^i_{DG}. Our proposed multinumerics scheme uses a finite volume method (FV) on the union of Ω^i_{FV}, denoted by Ω_F, and a discontinuous Galerkin (DG) method on the union of Ω^i_{DG}, denoted by Ω_D. Let $\mathscr{E}^h_D$ (resp. $\mathscr{E}^h_F$) be a subdivision of Ω_D (resp. Ω_F) made of cells V (Voronoi cells in Ω_F and either triangles/tetrahedra/hexaedra or Voronoi cells in Ω_D). We also denote by h_F (resp. h_D) the maximum diameter over all cells in Ω_F (resp. Ω_D) and we let $h = \max(h_F, h_D)$. We assume that the meshes match at the interface Γ_{DF} defined as:

$$\Gamma_{\mathrm{DF}} = \cup_i (\partial\Omega^i_{DG} \cap \partial\Omega^i_{FV})$$

The definition of the mesh $\mathscr{E}^h_F$ requires further notation. It is assumed that $\mathscr{E}^h_F$ is an admissible finite volume mesh, in the following sense:
(i) There is a family of nodes $\{x_V : V \in \mathscr{E}^h_F\}$ such that x_V belongs to $\overline{V}$ and if a face γ is shared by two neighboring cells V and W, it is assumed that x_W and x_V are distinct, and that the straight line going through x_V and x_W is orthogonal to γ.
(ii) For any boundary face $\gamma = \partial V \cap \partial\Omega$ for some V in $\mathscr{E}^h_F$, it is assumed that x_V does not lie on γ. However this condition can be relaxed.
We denote by $\Gamma_{\mathrm{F}}^{h,\mathscr{I}}$ the set of faces that belong to the interior of Ω_F and by $\Gamma_{\mathrm{F}}^{h,\partial}$ the set of boundary faces that belong to $\cup_i(\partial\Omega^i_{FV} \cap \partial\Omega)$. Similarly, the sets of interior and boundary faces of Ω_D are denoted by $\Gamma_{\mathrm{D}}^{h,\mathscr{I}}$ and $\Gamma_{\mathrm{D}}^{h,\partial}$ respectively. We also define $\Gamma_{\mathrm{F}}^h = \Gamma_{\mathrm{F}}^{h,\mathscr{I}} \cup \Gamma_{\mathrm{F}}^{h,\partial}$ and $\Gamma_{\mathrm{D}}^h = \Gamma_{\mathrm{D}}^{h,\mathscr{I}} \cup \Gamma_{\mathrm{D}}^{h,\partial}$. There remains the set of faces that belong to the interface Γ_{DF}; this particular set will be denoted by Γ_{DF}^h. We further decompose the boundary of Ω into inflow and outflow boundaries. The unit normal vector outward of Ω is denoted by n.

$$\Gamma_{\mathrm{D}}^{h,\partial-} = \{x \in \Gamma_{\mathrm{D}}^{h,\partial}, \quad \beta \cdot n \le 0\}, \quad \Gamma_{\mathrm{D}}^{h,\partial+} = \Gamma_{\mathrm{D}}^{h,\partial} \setminus \Gamma_{\mathrm{D}}^{h,\partial-}.$$

$$\Gamma_{\mathrm{F}}^{h,\partial-} = \{x \in \Gamma_{\mathrm{F}}^{h,\partial}, \quad \beta \cdot n \le 0\}, \quad \Gamma_{\mathrm{F}}^{h,\partial+} = \Gamma_{\mathrm{F}}^{h,\partial} \setminus \Gamma_{\mathrm{F}}^{h,\partial-}.$$

We now define a parameter d_γ that is associated to each face γ in $\Gamma_{\mathrm{F}}^h \cup \Gamma_{\mathrm{DF}}^h$. If the face γ is an interior face shared by two cells V and W in $\mathscr{E}^h_F$, the parameter d_γ is the Euclidean distance between the nodes x_V and x_W: $d_\gamma = d(x_V, x_W)$. If the face γ is a boundary face ($\gamma \subset \partial V \cap \partial\Omega$), the parameter d_γ is the distance between the node

x_V and the face γ, in other words $d_\gamma = d(x_V, y_\gamma)$, where y_γ denotes the non-empty intersection between the straight line going through x_V and orthogonal to γ. Finally, if the face γ lies on the interface Γ_{DF} and is shared by a cell V in $\mathscr{E}_F^h$ and a cell W in $\mathscr{E}_D^h$, the parameter d_γ is defined to be the distance between the node x_V and the edge γ. As in the boundary case, we can denote by y_γ the intersection between the straight line going through x_V and perpendicular to γ. Then, we have $d_\gamma = d(x_V, y_\gamma)$.

An admissible mesh in the finite volume regions is such that there is some positive number $\theta > 0$ such that

$$\begin{aligned} d_\gamma \geq \theta \max(h_V, h_W), \quad \forall \gamma \in \Gamma_{\mathrm{F}}^{h,\mathscr{I}}, \quad \gamma = \partial V \cap \partial W, \\ d_\gamma \geq \theta h_V, \quad \forall \gamma \in \Gamma_{\mathrm{F}}^{h,\partial}, \quad \gamma = \partial V \cap \partial \Omega, \\ d_\gamma \geq \theta h_V, \quad \forall \gamma \in \Gamma_{\mathrm{DF}}^{h}, \quad \gamma = \partial V \cap \partial W, \quad V \in \mathscr{E}_F^h, W \in \mathscr{E}_D^h. \end{aligned}$$

A standard harmonic average of the diffusion coefficient K is now defined:

$$\begin{aligned} K_\gamma = d_\gamma \left| \int_{x_V}^{x_W} \frac{ds}{K(s)} \right|^{-1}, \quad \forall \gamma \in \Gamma_{\mathrm{F}}^{h,\mathscr{I}}, \quad \gamma = \partial V \cap \partial W, \\ K_\gamma = d_\gamma \left| \int_{x_V}^{y_\gamma} \frac{ds}{K(s)} \right|^{-1}, \quad \forall \gamma \in \Gamma_{\mathrm{F}}^{h,\partial}, \quad \gamma = \partial V \cap \partial \Omega, \\ K_\gamma = d_\gamma \left| \int_{x_V}^{y_\gamma} \frac{ds}{K(s)} \right|^{-1}, \quad \forall \gamma \in \Gamma_{\mathrm{DF}}^{h}, \quad \gamma = \partial V \cap \partial W, \quad V \in \mathscr{E}_F^h, W \in \mathscr{E}_D^h. \end{aligned}$$

It is easy to see that K_γ is bounded above and below by k_1 and k_0 respectively. We denote by $|\gamma|$ the measure of the face γ.

Let X^{DG} be the space of discontinuous piecewise polynomials of degree $r \geq 1$ in the DG subdomains. Let X^{FV} be the space of piecewise constants in the FV subdomains. The restriction of the numerical solution to the DG subdomains (resp. FV subdomains) is denoted by u_{DG} (resp. u_{FV}).

2.1 Bilinear Forms

The differential operators are discretized by an interior penalty discontinuous Galerkin method in some subdomains and by a cell-centered finite volume method in other subdomains.

First, we define the jump of any discontinuous piecewise polynomial function. For any face γ, we fix a unit normal vector n_γ to γ. If γ is a boundary face, then n_γ is the outward normal to Ω. If γ belongs to the interface Γ_{DF}^h, then the vector n_γ is chosen to point from the DG region into the FV region. In the definition of the jump $[v]$ of a function v given below, we assume that the face γ is shared by two cells V and W, and that the normal vector n_γ points from V into W. For the interior faces, we define

$$[v]|_\gamma = v|_V - v|_W, \quad \gamma \in \Gamma_{\mathrm{F}}^{h,\mathscr{I}} \cup \Gamma_{\mathrm{D}}^{h,\mathscr{I}}, \quad \gamma = \partial V \cap \partial W$$

For the boundary faces, we define

$$[v]|_\gamma = v|_V, \quad \gamma \in \Gamma_{\mathrm{F}}^{h,\partial} \cup \Gamma_{\mathrm{D}}^{h,\partial}, \quad \gamma = \partial V \cap \partial \Omega.$$

In the definitions above, it is understood that $v|_W = v(x_W)$ if W is a cell in the FV subdomains.

The average of a discontinuous function v on a face is denoted by $\{v\}$ and defined below:

$$\begin{aligned} \{v\}|_\gamma &= \frac{1}{2}(v|_V + v|_W), \quad \forall \gamma = \partial V \cap \partial W, \\ \{v\}|_\gamma &= v|_V, \quad \forall \gamma = \partial V \cap \partial \Omega. \end{aligned}$$

Finally we define the upwind $v^\uparrow$ on the faces. For a given face γ in $\Gamma_{\mathrm{D}}^h \cup \Gamma_{\mathrm{F}}^h \cup \Gamma_{\mathrm{DF}}^h$ shared by cells V and W such that n_γ points from V into W, we have

$$v^\uparrow = \begin{cases} v|_V & \text{if} \quad \beta \cdot n_\gamma \geq 0, \\ v|_W & \text{if} \quad \beta \cdot n_\gamma < 0. \end{cases}$$

In what follows, we derive the bilinear forms corresponding to each subdomain. First, we multiply (1) by a function $v \in X^{DG}$, integrate over one DG cell V:

$$\int_V (K\nabla u - \beta u) \cdot \nabla v - \int_{\partial V} (K\nabla u - \beta u) \cdot n_V v = \int_V f v$$

We sum over all the cells in all the DG subdomains, use the definition of the normal vector n_γ and the regularity of the exact solution to obtain:

$$\begin{aligned} \sum_{V \in \mathscr{O}_D^h} \int_V (K\nabla u - \beta u) \cdot \nabla v - \sum_{\gamma \in \Gamma_{\mathrm{D}}^{h,\mathscr{I}}} \int_\gamma (\{K\nabla u\} - \beta u^\uparrow) \cdot n_\gamma [v] \\ - \sum_{\gamma \in \Gamma_{\mathrm{D}}^{h,\partial} \cup \Gamma_{\mathrm{DF}}^h} \int_\gamma (K\nabla u - \beta u) \cdot n_\gamma v = \sum_{V \in \mathscr{O}_D^h} \int_V f v \end{aligned}$$

Stabilization terms are added for the interior penalty discontinuous Galerkin method. The penalty parameter is denoted by $\sigma > 0$ and the symmetrization parameter by $\varepsilon \in \{-1, +1\}$. The penalty parameter is assumed to be large enough if $\varepsilon = -1$ and is taken equal to 1 if $\varepsilon = +1$. The parameter h_γ denotes the maximum diameter of the neighboring cells V and W, that share the face γ.

$$\begin{aligned} \sum_{V \in \mathscr{O}_D^h} \int_V (K\nabla u - \beta u) \cdot \nabla v - \sum_{\gamma \in \Gamma_{\mathrm{D}}^h} \int_\gamma \big(\{K\nabla u \cdot n_\gamma\}[v] - \varepsilon \{K\nabla v \cdot n_\gamma\}[u] \big) \\ + \sum_{\gamma \in \Gamma_{\mathrm{D}}^h} \sigma h_\gamma^{-1} \int_\gamma [u][v] + \sum_{\gamma \in \Gamma_{\mathrm{D}}^{h,\mathscr{I}}} \int_\gamma \beta \cdot n_\gamma u^\uparrow [v] + \sum_{\gamma \in \Gamma_{\mathrm{D}}^{h,\partial}} \int_\gamma \beta \cdot n_\gamma u v \end{aligned}$$

$$-\sum_{\gamma\in\Gamma_{\mathrm{DF}}^h}\int_\gamma(K\nabla u-\beta u)\cdot n_\gamma v=\sum_{V\in\mathscr{E}_D^h}\int_V fv+\varepsilon\sum_{\gamma\in\Gamma_{\mathrm{D}}^{h,\partial}}\int_\gamma\{K\nabla v\cdot n_\gamma\}g + \sum_{\gamma\in\Gamma_{\mathrm{D}}^{h,\partial}}\sigma h_\gamma^{-1}\int_\gamma gv$$

From this derivation, we define the bilinear form for the DG subdomains as:

$$a_{\mathrm{DG}}(u,v)=\sum_{V\in\mathscr{E}_D^h}\int_V(K\nabla u-\beta u)\cdot\nabla v-\sum_{\gamma\in\Gamma_{\mathrm{D}}^h}\int_\gamma\big(\{K\nabla u\cdot n_\gamma\}[v]-\varepsilon\{K\nabla v\cdot n_\gamma\}[u]\big) + \sum_{\gamma\in\Gamma_{\mathrm{D}}^h}\sigma h_\gamma^{-1}\int_\gamma[u][v]+\sum_{\gamma\in\Gamma_{\mathrm{D}}^{h,\mathscr{I}}}\int_\gamma\beta\cdot n_\gamma u^{\uparrow}[v]+\sum_{\gamma\in\Gamma_{\mathrm{D}}^{h,\partial+}}\int_\gamma\beta\cdot n_\gamma uv + \sum_{\gamma\in\Gamma_{\mathrm{DF}}^h}\frac{|\gamma|}{d_\gamma}K_\gamma u(y_\gamma)v(y_\gamma)+\sum_{\gamma\in\Gamma_{\mathrm{DF}}^h}\int_{\gamma+}\beta\cdot n_\gamma uv$$

In the last term, the subset of a face γ on which $\beta\cdot n_\gamma$ is non-negative is denoted by $\gamma+$. This corresponds to the outflow part of the face. The inflow part is denoted by $\gamma-$.

Second, we multiply (1) by a function $v\in X^{FV}$, that is piecewise constant, integrate over one FV cell V:

$$-\int_{\partial V}(K\nabla u-\beta u)\cdot n_V v=\int_V fv$$

We sum over all the FV cells and use the regularity of the exact solution:

$$\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\mathscr{I}}}\int_\gamma(-K\nabla u\cdot n_\gamma+\beta\cdot n_\gamma u^{\uparrow})[v]+\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\partial}}\int_\gamma(-K\nabla u\cdot n_\gamma+\beta\cdot n_\gamma u)v + \sum_{\gamma\in\Gamma_{\mathrm{DF}}^h}\int_\gamma(K\nabla u-\beta u)\cdot n_\gamma v=\sum_{V\in\mathscr{E}_F^h}\int_V fv$$

A cell-centered finite difference approximation is used to approximate the flux across the faces. Therefore we define the bilinear form in the FV regions as:

$$a_{\mathrm{FV}}(u,v)=\sum_{\gamma\in\Gamma_{\mathrm{F}}^h}\frac{|\gamma|}{d_\gamma}K_\gamma[u][v]+\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\mathscr{I}}}\int_\gamma\beta\cdot n_\gamma u^{\uparrow}[v]+\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\partial+}}\int_\gamma\beta\cdot n_\gamma uv + \sum_{\gamma\in\Gamma_{\mathrm{DF}}^h}\frac{|\gamma|}{d_\gamma}K_\gamma uv-\sum_{\gamma\in\Gamma_{\mathrm{DF}}^h}\int_{\gamma-}\beta\cdot n_\gamma uv$$

Finally the source function f and the boundary conditions are handled by the following bilinear forms:

$$\ell_{DG}(v) = \int_{\Omega_D} fv + \varepsilon \sum_{\gamma\in\Gamma_{\rm D}^{h,\partial}} \int_\gamma K\nabla v\cdot n_\gamma g + \sum_{\gamma\in\Gamma_{\rm D}^{h,\partial}} \sigma h_\gamma^{-1}\int_\gamma gv - \sum_{\gamma\in\Gamma_{\rm D}^{h,\partial-}}\int_\gamma \beta\cdot n_\gamma gv$$

$$\ell_{FV}(v) = \int_{\Omega_F} fv + \sum_{\gamma\in\Gamma_{\rm F}^{h,\partial}} \frac{|\gamma|}{d_\gamma} K_\gamma g(y_\gamma)v - \sum_{\gamma\in\Gamma_{\rm F}^{h,\partial-}}\int_\gamma \beta\cdot n_\gamma gv.$$

2.2 A monolithic formulation

The definition of the multinumeric scheme, without Lagrange multipliers, is given in this section. Existence and uniqueness of the solution is shown.

The numerical method is as follows: find $u_{DG}\in X^{DG}$, $u_{FV}\in X^{FV}$ such that

$$a_{DG}(u_{DG},v_{DG}) = \ell_{DG}(v_{DG}) + \sum_{\gamma\in\Gamma_{\rm DF}^h} \frac{|\gamma|}{d_\gamma} K_\gamma u_{FV} v_{DG}(y_\gamma) - \sum_{\gamma\in\Gamma_{\rm DF}^h}\int_{\gamma-} \beta\cdot n_\gamma u_{FV} v_{DG}, \quad (3)$$

$$a_{FV}(u_{FV},v_{FV}) = \ell_{FV}(v_{FV}) + \sum_{\gamma\in\Gamma_{\rm DF}^h} \frac{|\gamma|}{d_\gamma} K_\gamma u_{DG}(y_\gamma) v_{FV} + \sum_{\gamma\in\Gamma_{\rm DF}^h}\int_{\gamma+} \beta\cdot n_\gamma u_{DG} v_{FV}, \quad (4)$$

for all $v_{DG}\in X^{DG}$ and all $v_{FV}\in X^{FV}$.

Lemma 1. *There exists a unique solution (u_{DG},u_{FV}), satisfying (3)-(4).*

Proof. Let us assume that $f=g=0$ and take $v_{DG}=u_{DG}$ and $v_{FV}=u_{FV}$ in (3)-(4). We have

$$a_{DG}(u_{DG},u_{DG}) + a_{FV}(u_{FV},u_{FV}) = 2\sum_{\gamma\in\Gamma_{\rm DF}^h} \frac{|\gamma|}{d_\gamma} K_\gamma u_{FV} u_{DG}(y_\gamma) + \sum_{\gamma\in\Gamma_{\rm DF}^h}\int_\gamma |\beta\cdot n_\gamma| u_{DG} u_{FV}.$$

We expand the DG form:

$$a_{\rm DG}(u_{DG},u_{DG}) = \sum_{V\in\mathscr{E}_D^h} \|K^{1/2}\nabla u_{DG}\|^2_{L^2(V)} + \sum_{\gamma\in\Gamma_{\rm D}^h} \sigma h_\gamma^{-1} \|[u_{DG}]\|^2_{L^2(\gamma)}$$

$$+ \sum_{\gamma\in\Gamma_{\rm DF}^h} \frac{|\gamma|}{d_\gamma} K_\gamma u_{DG}(y_\gamma)^2 - (1-\varepsilon)\sum_{\gamma\in\Gamma_{\rm D}^h}\int_\gamma \{K\nabla u_{DG}\cdot n_\gamma\}[u_{DG}] - \sum_{V\in\mathscr{E}_D^h}\int_V \beta u_{DG}\cdot\nabla u_{DG}$$

$$+ \sum_{\gamma\in\Gamma_{\rm D}^{h,\mathscr{I}}}\int_\gamma \beta\cdot n_\gamma u_{DG}^{\uparrow}[u_{DG}] + \sum_{\gamma\in\Gamma_{\rm D}^{h,\partial+}}\int_\gamma \beta\cdot n_\gamma u_{DG}^2 + \sum_{\gamma\in\Gamma_{\rm DF}^h}\int_{\gamma+} \beta\cdot n_\gamma u_{DG}^2$$

Using standard techniques to DG methods [9], one can show that

$$-\sum_{V\in\mathscr{E}_D^h}\int_V \beta u_{DG}\cdot\nabla u_{DG} + \sum_{\gamma\in\Gamma_{\rm D}^{h,\mathscr{I}}}\int_\gamma \beta\cdot n_\gamma u_{DG}^{\uparrow}[u_{DG}] + \sum_{\gamma\in\Gamma_{\rm D}^{h,\partial+}}\int_\gamma \beta\cdot n_\gamma u_{DG}^2$$

$$= \frac{1}{2}\sum_{\gamma\in\Gamma_{\mathrm{D}}^h} \| |\boldsymbol{\beta}\cdot n_\gamma|^{1/2}[u_{DG}]\|^2_{L^2(\gamma)} - \frac{1}{2}\sum_{\gamma\in\Gamma_{\mathrm{DF}}^h}\int_\gamma \boldsymbol{\beta}\cdot n_\gamma u_{DG}^2$$

In addition, we can show that there is a constant $M>0$ independent of h such that

$$\sum_{V\in\mathscr{O}_D^h} \|K^{1/2}\nabla u_{DG}\|^2_{L^2(V)} + \sum_{\gamma\in\Gamma_{\mathrm{D}}^h}\sigma h_\gamma^{-1}\|[u_{DG}]\|^2_{L^2(\gamma)} - (1-\varepsilon)\sum_{\gamma\in\Gamma_{\mathrm{D}}^h}\int_\gamma \{K\nabla u_{DG}\cdot n_\gamma\}[u_{DG}]$$
$$\geq M\left(\sum_{V\in\mathscr{O}_D^h} \|K^{1/2}\nabla u_{DG}\|^2_{L^2(V)} + \sum_{\gamma\in\Gamma_{\mathrm{D}}^h}\sigma h_\gamma^{-1}\|[u_{DG}]\|^2_{L^2(\gamma)}\right)$$

For the FV bilinear form, we have

$$a_{\mathrm{FV}}(u_{FV},u_{FV}) = \sum_{\gamma\in\Gamma_{\mathrm{F}}^h}\frac{|\gamma|}{d_\gamma}K_\gamma[u_{FV}]^2 + \sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\mathscr{I}}}\int_\gamma \boldsymbol{\beta}\cdot n_\gamma u_{FV}^{\uparrow}[u_{FV}] + \sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\partial+}}\int_\gamma \boldsymbol{\beta}\cdot n_\gamma u_{FV}^2$$
$$+ \sum_{\gamma\in\Gamma_{\mathrm{DF}}^h}\frac{|\gamma|}{d_\gamma}K_\gamma u_{FV}^2 - \sum_{\gamma\in\Gamma_{\mathrm{DF}}^h}\int_{\gamma-}\boldsymbol{\beta}\cdot n_\gamma u_{FV}^2$$

We observe that, if $u_{FV}^{\downarrow}$ denotes the downwind value of u_{FV}, we have

$$\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\mathscr{I}}}\int_\gamma \boldsymbol{\beta}\cdot n_\gamma u_{FV}^{\uparrow}[u_{FV}] = \frac{1}{2}\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\mathscr{I}}}\| |\boldsymbol{\beta}\cdot n_\gamma|^{1/2}[u_{FV}]\|^2_{L^2(\gamma)}$$
$$+\frac{1}{2}\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\mathscr{I}}}\int_\gamma |\boldsymbol{\beta}\cdot n_\gamma|((u_{FV}^{\uparrow})^2-(u_{FV}^{\downarrow})^2)$$

Since $\boldsymbol{\beta}$ is divergence-free, we obtain

$$\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\mathscr{I}}}\int_\gamma \boldsymbol{\beta}\cdot n_\gamma u_{FV}^{\uparrow}[u_{FV}] = \frac{1}{2}\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\mathscr{I}}}\| |\boldsymbol{\beta}\cdot n_\gamma|^{1/2}[u_{FV}]\|^2_{L^2(\gamma)}$$
$$-\frac{1}{2}\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\partial}}\int_\gamma \boldsymbol{\beta}\cdot n_\gamma u_{FV}^2 + \frac{1}{2}\sum_{\gamma\in\Gamma_{\mathrm{DF}}^h}\int_\gamma \boldsymbol{\beta}\cdot n_\gamma u_{FV}^2$$

Combining the results above yields

$$M\sum_{V\in\mathscr{O}_D^h} \|K^{1/2}\nabla u_{DG}\|^2_{L^2(V)} + M\sum_{\gamma\in\Gamma_{\mathrm{D}}^h}\sigma h_\gamma^{-1}\|[u_{DG}]\|^2_{L^2(\gamma)} + \frac{1}{2}\sum_{\gamma\in\Gamma_{\mathrm{D}}^h}\| |\boldsymbol{\beta}\cdot n_\gamma|^{1/2}[u_{DG}]\|_{L^2(\gamma)}$$
$$+\sum_{\gamma\in\Gamma_{\mathrm{F}}^h}\frac{|\gamma|}{d_\gamma}K_\gamma[u_{FV}]^2 + \frac{1}{2}\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\mathscr{I}}}\| |\boldsymbol{\beta}\cdot n_\gamma|^{1/2}[u_{FV}]\|^2_{L^2(\gamma)} + \frac{1}{2}\sum_{\gamma\in\Gamma_{\mathrm{F}}^{h,\partial}}\int_\gamma |\boldsymbol{\beta}\cdot n_\gamma| u_{FV}^2$$

$$+\sum_{\gamma\in\Gamma_{\rm DF}^h}\frac{|\gamma|}{d_\gamma}K_\gamma(u_{DG}(y_\gamma)-u_{FV})^2+\frac{1}{2}\sum_{\gamma\in\Gamma_{\rm DF}^h}\int_\gamma|\beta\cdot n_\gamma|(u_{DG}-u_{FV})^2\le 0$$

The inequality above immediately implies that u_{DG} and u_{FV} are zero everywhere. Thus, we have proved uniqueness of the solution. Since the finite-dimensional problem is linear, this is equivalent to showing existence of the solution.

2.3 Formulation with Lagrange multipliers

In this section, we rewrite the method (3)-(4) in a hybridized form for the elliptic problem. Lagrange multipliers are defined on the interface between the subdomains.

Let $\Lambda_h^0\subset L^2(\Gamma_{12})$ be the finite dimensional space of piecewise constants on the partition of Γ_{12}. Assume that the convection vector β is zero. The hybridized DG-FV scheme becomes: solve for $u_{\rm DG}\in X^{\rm DG}$, $u_{\rm FV}\in X^{\rm FV}$, $\lambda_{DG}\in\Lambda_h^0$, $\lambda_{FV}\in\Lambda_h^0$ satisfying

$$a_{DG}(u_{DG},v_{DG})=\ell_{DG}(v_{DG})+\sum_{\gamma\in\Gamma_{\rm DF}^h}\frac{|\gamma|}{d_\gamma}K_\gamma\lambda_{FV}v_{DG}(y_\gamma),\quad\forall v\in X^{DG}\tag{5}$$

$$a_{FV}(u_{FV},v_{FV})=\ell_{FV}(v_{FV})+\sum_{\gamma\in\Gamma_{\rm DF}^h}\frac{|\gamma|}{d_\gamma}K_\gamma\lambda_{DG}v_{FV},\quad\forall v\in X^{FV}\tag{6}$$

$$\sum_{\gamma\in\Gamma_{\rm DF}^h}\int_\gamma(\lambda_{DG}-u_{DG}(y_\gamma))\mu=0,\quad\forall\mu\in\Lambda_h^0\tag{7}$$

$$\sum_{\gamma\in\Gamma_{\rm DF}^h}\int_\gamma(\lambda_{FV}-u_{FV})\mu=0,\quad\forall\mu\in\Lambda_h^0\tag{8}$$

Lemma 2. *There exists a unique solution to (5)-(8)*

Proof. To show uniqueness of the solution, we assume that $f=g=0$ and take $v_{DG}=u_{DG}$ and $v_{FV}=u_{FV}$ in (5) and (6). We observe that (7) and (8) imply that

$$\lambda_{DG}|_\gamma=u_{DG}(y_\gamma),\quad\lambda_{FV}|_\gamma=u_{FV},\quad\forall\gamma\in\Gamma_{\rm DF}^h$$

The rest of the proof follows the proof of Lemma 1.

3 Error analysis

In this section, convergence of the multinumeric approach is shown under some regularity assumptions of the exact solution.

Assume that the relative gradient of the exact solution near the interfaces with respect to the gradient in the DG subdomains is small. In particular, given a face

$\gamma \in \Gamma_{\mathrm{DF}}^h$ that belongs to a DG cell denoted by V_γ, assume that there is a constant C independent of h_D such that

$$\Big(\sum_{\gamma \in \Gamma_{\mathrm{DF}}^h} \|\nabla u\|^2_{L^2(V_\gamma)}\Big)^{1/2} \leq C h_D \Big(\sum_{V \in \mathscr{O}_D^h} \|\nabla u\|^2_{L^2(V)}\Big)^{1/2} \tag{9}$$

This assumption is an indicator on how to choose the interface. We want to place the interface where the exact solution does not vary as much as it does in the interior of the discontinuous Galerkin domain. In the simple case where the exact solution is linear and its gradient is uniformly constant, this assumption is not satisfied (see remark 1).

We recall that by convention, the jump $[u - u_{FV}]$ on an edge that belongs to Γ_{F}^h is the difference between $u(x_V) - u_{FV}(x_V)$ and $u(x_W) - u_{FV}(x_W)$ if the edge is shared by the Voronoi cells V and W.

Theorem 1. *Assume that u belongs to $H^2(\Omega)$ and that $u|_{\Omega_D}$ belongs to $H^{r+1}(\mathscr{O}_D^h)$, for $r \geq 1$. Under the assumption (9), there exists a constant C independent of h such that*

$$\begin{aligned}\sum_{V \in \mathscr{O}_D^h} \|K^{1/2}\nabla(u - u_{DG})\|^2_{L^2(V)} + \sum_{\gamma \in \Gamma_{\mathrm{D}}^h} \sigma h_\gamma^{-1} \|[u - u_{DG}]\|^2_{L^2(\gamma)} + \sum_{\gamma \in \Gamma_{\mathrm{F}}^h} \frac{\gamma}{d_\gamma} K_\gamma [u - u_{FV}]^2 \\ + \sum_{\gamma \in \Gamma_{\mathrm{D}}^h} \| |\beta \cdot n_\gamma|^{1/2} [u - u_{DG}]\|^2_{L^2(\gamma)} + \sum_{\gamma \in \Gamma_{\mathrm{F}}^h} \| |\beta \cdot n_\gamma|^{1/2} [u - u_{DG}]\|^2_{L^2(\gamma)} \\ + \sum_{\gamma \in \Gamma_{\mathrm{DF}}^h} \frac{|\gamma|}{d_\gamma} K_\gamma (u_{DG}(y_\gamma) - u_{FV})^2 \leq C(h_D^2 + h_F^2)\end{aligned}$$

Proof. An outline of the proof is given. First we observe that the scheme (3)-(4) is not consistent because of the use of finite difference approximations in the FV subdomains and on the interfaces between the subdomains. We introduce an optimal approximation, $\tilde{u}$, of the exact solution such that $\tilde{u}|_{\Omega_D}$ (resp. $\tilde{u}|_{\Omega_F}$) belongs to X^{DG} (resp. X^{FV}). We define

$$\chi_{DG} = u_{DG} - \tilde{u}|_{\Omega_D}, \quad \chi_{FV} = u_{FV} - \tilde{u}|_{\Omega_F}, \quad \xi = u - \tilde{u}$$

An error equation can be obtained:

$$\begin{aligned}a_{DG}(\chi_{DG}, v_{DG}) + a_{FV}(\chi_{FV}, v_{FV}) + \sum_{\gamma \in \Gamma_{\mathrm{DF}}^h} \int_{\gamma_-} \beta \cdot n_\gamma \chi_{FV} v_{DG} - \sum_{\gamma \in \Gamma_{\mathrm{DF}}^h} \frac{|\gamma|}{d_\gamma} K_\gamma \chi_{FV} v_{DG}(y_\gamma) \\ - \sum_{\gamma \in \Gamma_{\mathrm{DF}}^h} \int_{\gamma_+} \beta \cdot n_\gamma \chi_{DG} v_{FV} - \sum_{\gamma \in \Gamma_{\mathrm{DF}}^h} \frac{|\gamma|}{d_\gamma} K_\gamma \chi_{DG}(y_\gamma) v_{FV} = a_{DG}(\xi_{DG}, v_{DG}) + a_{FV}(\xi_{FV}, v_{FV}) \\ + \sum_{\gamma \in \Gamma_{\mathrm{DF}}^h} \int_{\gamma_-} \beta \cdot n_\gamma \xi|_{\Omega_F} v_{DG} - \sum_{\gamma \in \Gamma_{\mathrm{DF}}^h} \frac{|\gamma|}{d_\gamma} K_\gamma \xi|_{\Omega_F} v_{DG}(y_\gamma) - \sum_{\gamma \in \Gamma_{\mathrm{DF}}^h} \int_{\gamma_+} \beta \cdot n_\gamma \xi|_{\Omega_D} v_{FV}\end{aligned}$$

$$- \sum_{\gamma \in \Gamma_{\rm DF}^h} \frac{|\gamma|}{d_\gamma} K_\gamma \xi|_{\Omega_D}(y_\gamma) v_{FV} + R,$$

where R is a residual term resulting from the consistency error. An expression for R is:

$$R = \sum_{\gamma \in \Gamma_{\rm F}^h} R_\gamma(u)[v_{FV}] + \sum_{\gamma \in \Gamma_{\rm DF}^h} \int_\gamma K \nabla u \cdot n_\gamma (v_{DG} - v_{DG}(y_\gamma)) + \sum_{\gamma \in \Gamma_{\rm DF}^h} \int_\gamma R_\gamma(u)(v_{DG}(y_\gamma) - v_{FV}) \tag{10}$$

The residual quantities $R_\gamma(u)$ are defined on the interior faces of the FV subdomains as

$$R_\gamma(u) = -\int_\gamma K \nabla u \cdot n_\gamma - \frac{|\gamma|}{d_\gamma} K_\gamma (u(x_V) - u(x_W)), \quad \forall \gamma = \partial V \cap \partial W \quad \forall \gamma \in \Gamma_{\rm F}^{h,\mathscr{I}}$$

This expression is slightly modified for the exterior boundary faces of the FV subdomains:

$$R_\gamma(u) = -\int_\gamma K \nabla u \cdot n_\gamma - \frac{|\gamma|}{d_\gamma} K_\gamma (u(x_V) - g(y_\gamma)), \quad \forall \gamma = \partial V, \quad \forall \gamma \in \Gamma_{\rm F}^{h,\partial}$$

For the interfaces between the FV and DG subdomains, the residual term is defined as

$$R_\gamma(u) = -K \nabla u \cdot n_\gamma - \frac{K_\gamma}{d_\gamma}(u(y_\gamma) - u(x_W)), \quad \forall \gamma \in \partial W, W \in \mathscr{E}_F^h, \quad \forall \gamma \in \Gamma_{\rm DF}^h$$

Next, we choose $v_{DG} = \chi_{DG}$ and $v_{FV} = \chi_{FV}$ in the error equation. The error estimate follows by using trace inequalities, approximation results, and the following bounds on the residuals, that involve the Hessian matrix $\mathbf{H}(u)$ (see [3]):

$$|R_\gamma(u)|^2 \le C \frac{h_F^2 |\gamma|}{d_\gamma} \int_{\mathscr{V}_\gamma} |\mathbf{H}(u)|^2, \quad \forall \gamma \in \Gamma_{\rm F}^h$$

$$\left(\int_\gamma |R_\gamma(u)| \right)^2 \le C \frac{h_F^2 |\gamma|}{d_\gamma} \int_{\mathscr{V}_\gamma} |\mathbf{H}(u)|^2, \quad \forall \gamma \in \Gamma_{\rm DF}^h$$

The Hessian is integrated over the region $\mathscr{V}_\gamma$ defined by

$$\mathscr{V}_\gamma = \mathscr{V}_{W,\gamma} \cup \mathscr{V}_{V,\gamma}, \quad \forall \gamma = \partial V \cap \partial W$$

with

$$\mathscr{V}_{W,\gamma} = \{ t x_W + (1-t)x : x \in \gamma, t \in [0,1] \}$$

Remark 1. If the assumption (9) is removed, the multinumeric approach converges suboptimally. Indeed, there is a loss of $h_D^{1/2}$ in the bound of the last term in the definition of the residual in (10).

4 Conclusions

Cell-centered finite volume methods use Voronoi cells for unstructured meshes. Discontinuous Galerkin methods converge on general mesh elements including Voronoi grids. In addition, for two-dimensional problems, Voronoi cells can naturally and easily be partitioned into triangles by using the underlying Delaunay triangulation. In this work, we formulate and analyze a method that couples DG and FV methods via mesh interfaces. One appealing feature of the method is that, once a Voronoi grid is built, the decomposition of the domain into DG regions and FV regions is done easily and this decomposition can vary with each simulation.

Acknowledgements This work was partially funded by NSF-DMS and NHARP.

References

1. Brezzi, F., Johnson, C.: On the coupling of boundary integral and finite element methods. Calcolo **16**, 189–201 (1979)
2. Chidyagwai, P., Mishev, I., Riviere, B.: On the coupling of finite volume and discontinuous Galerkin method for elliptic problems. Journal of Computational and Applied Mathematics **231**, 2193–2204 (2011)
3. Eymard, R., Gallouët, T., Herbin, R.: Finite Volume Methods. Handbook of Numerical Analysis, vol. VII (2000)
4. Forsyth, P., Sammon, P.: Quadratic convergence for cell-centered grids. Appl. Numer. Math. **4**, 377–394 (1988)
5. Johnson, C., Nédélec, J.: On the coupling of boundary integral and finite element methods. Mathematics of Computation **35**, 269–284 (1980)
6. Lazarov, R., Mishev, I., Vassilevski, P.: Finite volume methods for convection-diffusion problems. SIAM J. Numer. Anal. pp. 31–55 (1996)
7. Lazarov, R., Pasciak, J., Vassilevski, P.: Coupling mixed and finite volume discretizations of convection-diffusion equations on non-matching grids. In: Proceedings: Finite Volumes for Complex Applications Hermes Science (1999)
8. Mishev, I.: Finite volume methods on Voronoi meshes. Numerical Methods for Partial Differential Equations **14**, 193–212 (1998)
9. Riviere, B.: Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation. SIAM (2008)
10. Riviere, B., Wheeler, M.: Coupling locally conservative methods for single phase flow. Computational Geosciences **6**, 269–284 (2002)
11. Riviere, B., Yang, X.: Convergence analysis of a coupled method for time-dependent convection-diffusion equations. Numerical Methods for Partial Differential Equations, **30**, 133–157 (2013), doi: 10.1002/num.21800.
12. Vassilevski, P., Petrova, S., Lazarov, R.: Finite difference schemes on triangular cell-centered grids with local refinement. SIAM J. Sci. Statist. Comput. **13**, 1287–1313 (1992)

3-D FETI-DP preconditioners for composite finite element-discontinuous Galerkin methods

Maksymilian Dryja[1] and Marcus Sarkis[2]

1 Introduction

In this paper a Nitsche-type discretization based on discontinuous Galerkin (DG) method for an elliptic three-dimensional problem with discontinuous coefficients is considered. The problem is posed on a polyhedral region Ω which is a union of N disjoint polyhedral subdomains Ω_i of diameter $O(H_i)$ and we assume that this partition is geometrically conforming. Inside each subdomain, a conforming finite element space on a quasiuniform triangulation with mesh size $O(h_i)$ is introduced. Large discontinuities on the coefficients and nonmatching meshes are allowed to occur only across $\partial\Omega_i$. In order to deal with the nonconformity of the FE spaces across subdomain interfaces, a discrete problem is formulated using the symmetric interior penalty DG method only on the subdomain interfaces. For solving the resulting discrete system, FETI-DP type of methods are designed and fully analyzed. This paper extends the 2-D results in [2] to 3-D problems.

2 Differential and discrete problems

Consider the following problem: *Find* $u^*_{ex} \in H^1_0(\Omega)$ *such that*

$$a(u^*_{ex}, v) = f(v) \quad \text{for all} \quad v \in H^1_0(\Omega), \tag{1}$$

where

$$a(u,v) := \sum_{i=1}^{N} \int_{\Omega_i} \rho_i(x) \nabla u \cdot \nabla v \, dx \quad \text{and} \quad f(v) := \int_{\Omega} f v \, dx.$$

To simplify the presentation, we assume that $\rho_i(x)$ is equal to positive constant ρ_i.

We now consider the discrete problem associated to (1). Let $X_i(\Omega_i)$ be the regular finite element (FE) space of piecewise linear and continuous functions in Ω_i and define

$$X(\Omega) = \prod_{i=1}^{N} X_i(\Omega_i) \equiv X_1(\Omega_1) \times X_2(\Omega_2) \times \cdots \times X_N(\Omega_N).$$

[1]Department of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland, e-mail: dryja@mimuw.edu.pl [2] Mathematical Sciences Department, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609, USA, and and Instituto de Matemática Pura e Aplicada - IMPA, Estrada Dona Castorina 110, CEP 22460-320, Rio de Janeiro, Brazil, e-mail: msarkis@wpi.edu

We note that we do not assume that the functions in $X_i(\Omega_i)$ vanish on $\partial\Omega_i \cap \partial\Omega$.

Let us denote $\bar{F}_{ij} := \partial\Omega_i \cap \partial\Omega_j$ as a face of $\partial\Omega_i$ and $\bar{F}_{ji} := \partial\Omega_j \cap \partial\Omega_i$ as a face of $\partial\Omega_j$. In spite of the common face F_{ij} and F_{ji} being geometrically the same, they will be treated separately since we consider different triangulations on $\bar{F}_{ij} \subset \partial\Omega_i$ with a mesh parameter h_i and on $\bar{F}_{ji} \subset \partial\Omega_j$ with a mesh parameter h_j. We denote the interior h_i-nodes of F_{ij} and the h_j-nodes of F_{ji} by F_{ijh} and F_{jih}, respectively.

Let us denote by $\mathscr{F}_i^0$ the set of indices j of Ω_j which has a common face F_{ji} with Ω_i. To take into account also of these faces of Ω_i which belong to $\partial\Omega$, we introduce a set of indices $\mathscr{F}_i^\partial$ to refer theses faces. The set of indices of all faces of Ω_i is denoted by $\mathscr{F}_i := \mathscr{F}_i^0 \cup \mathscr{F}_i^\partial$. A discrete problem is obtained by a composite FE/DG method, see [1], is of the form: *Find* $u^* = \{u_i^*\}_{i=1}^N \in X(\Omega)$ *where* $u_i \in X_i(\Omega_i)$, *such that*

$$a_h(u^*, v) = f(v) \quad \text{for all} \quad v = \{v_i\}_{i=1}^N \in X(\Omega), \tag{2}$$

where

$$a_h(u,v) := \sum_{i=1}^N a_i'(u,v), \quad f(v) := \sum_{i=1}^N \int_{\Omega_i} f v_i \, dx,$$

$$a_i'(u,v) := \{a_i(u,v) + p_i(u,v)\} + s_i(u,v) \equiv \{d_i(u,v)\} + s_i(u,v), \tag{3}$$

where

$$a_i(u,v) := \int_{\Omega_i} \rho_i \nabla u_i \cdot \nabla v_i \, dx,$$

$$p_i(u,v) := \sum_{j \in \mathscr{F}_i} \int_{F_{ij}} \frac{\delta}{l_{ij}} \frac{\rho_i}{h_{ij}} (u_j - u_i)(v_j - v_i) \, ds,$$

and

$$s_i(u,v) := \sum_{j \in \mathscr{F}_i} \int_{F_{ij}} \frac{1}{l_{ij}} \left(\rho_i \frac{\partial u_i}{\partial n} (v_j - v_i) + \rho_i \frac{\partial v_i}{\partial n} (u_j - u_i) \right) ds.$$

Here, when $j \in \mathscr{F}_i^0$, we set $l_{ij} = 2$ and let $h_{ij} := 2h_i h_j/(h_i + h_j)$, i.e., the harmonic average of h_i and h_j. When $j \in \mathscr{F}_i^\partial$, we denote the boundary faces $F_{ij} \subset \partial\Omega_i$ by $F_{i\partial}$ and set $l_{i\partial} = 1$ and $h_{i\partial} = h_i$, and on the artificial face $F_{ji} \equiv F_{\partial i}$, we set $u_\partial = 0$ and $v_\partial = 0$. The partial derivative $\frac{\partial}{\partial n}$ denotes the outward normal derivative on $\partial\Omega_i$ and δ is the sufficiently large penalty parameter. For details on accuracy and well-posedness, see [2, 1] and references there in. In particular, we show that exists positive constants γ_0 and γ_1, which do not depend on the ρ_i, h_i and H_i, such that

$$\gamma_0 a_h(u,u) \le \sum_{i=1}^N d_i(u,u) \le \gamma_1 a_h(u,u) \quad \text{for all} \quad u \in X(\Omega).$$

3 Schur complement systems and discrete harmonic extensions

This section is similar to Section 3 in [2] with a few natural changes when passing from the 2-D to the 3-D case, and we refer to that for more details.

- Define the sets Ω_i', Γ_i, Γ_i', I_i, Γ, Γ', I and Ω' by

$$\Omega_i' = \overline{\Omega}_i \bigcup \{\cup_{j\in\mathscr{F}_i^0}\bar{F}_{ji}\}, \quad \Gamma_i = \overline{\partial\Omega_i\backslash\partial\Omega}, \quad \Gamma_i' = \Gamma_i \bigcup \{\cup_{j\in\mathscr{F}_i^0}\bar{F}_{ji}\},$$

$$\Gamma = \bigcup_{i=1}^N \Gamma_i, \quad \Gamma' = \prod_{i=1}^N \Gamma_i', \quad I_i = \Omega_i'\backslash\Gamma_i', \quad I = \prod_{i=1}^N I_i \quad \text{and} \quad \Omega' = \prod_{i=1}^N \Omega_i'.$$

- Define the space $W_i(\Omega_i')$ by

$$W_i(\Omega_i') = X_i(\Omega_i) \times \prod_{j\in\mathscr{F}_i^0} X_i(\bar{F}_{ji}), \text{ where } X_i(\bar{F}_{ji}) = X_j(\Omega_j)_{|\bar{F}_{ji}}.$$

  A function $u_i \in W_i(\Omega_i')$ will be represented as

$$u_i = \{(u_i)_i, \{(u_i)_j\}_{j\in\mathscr{F}_i^0}\},$$

  where $(u_i)_i := u_{i|\overline{\Omega}_i}$ (u_i restricted to $\overline{\Omega}_i$) and $(u_i)_j := u_{i|\bar{F}_{ji}}$ (u_i restricted to $\bar{F}_{ji}$).
- For the definition of the discrete harmonic extension operators $\mathscr{H}_i'$ and $\mathscr{H}_i$ (elimination of I_i variables) with respect to the bilinear forms a_i' and a_i, see [2].
- The matrices A_i' and S_i' are defined by

$$a_i'(u_i,v_i) = \langle A_i'u_i, v_i\rangle \;\; u_i,v_i \in W_i(\Omega_i'), \quad a_i'(u_i,v_i) = \langle S_i'u_i, v_i\rangle \;\; u_i,v_i \in W_i(\Gamma_i').$$

- $W_i(\Gamma_i') \subset W_i(\Omega_i')$ denotes the $\mathscr{H}_i'$-discrete harmonic functions.
- Define $W(\Omega') = \prod_{i=1}^N W_i(\Omega_i')$ and $W(\Gamma') = \prod_{i=1}^N W_i(\Gamma_i')$.
- Let the subspace $\hat{W}(\Omega') \subset W(\Omega')$ consist of functions $u = \{u_i\}_{i=1}^N \in W(\Omega')$ which are continuous on Γ, that is, for all $1 \le i \le N$ satisfy

$$(u_i)_i(x) = (u_j)_i(x) \quad \text{for all } x \in \bar{F}_{ij} \text{ for all } j \in \mathscr{F}_i^0$$

  and

$$(u_i)_j(x) = (u_j)_j(x) \quad \text{for all } x \in \bar{F}_{ji} \text{ for all } j \in \mathscr{F}_i^0.$$

  We note that $\hat{W}(\Omega')$ can be identified to $X(\Omega)$.
- $\hat{W}(\Gamma')$ denotes the subspace of $\hat{W}(\Omega')$ of $\mathscr{H}_i'$-discrete harmonic functions.
- The rest of Section 3 in [2] remains the same for 3-D problems. In particular, by eliminating the interior variables I from the system (2), we obtain

$$\hat{S}u_\Gamma^* = g_\Gamma. \tag{4}$$

We note that $\hat{S}$ can be assembled from S_i', i.e., $\hat{S} = \sum_{i=1}^N R_{\Gamma_i'}^T S_i' R_{\Gamma_i'}$, where $R_{\Gamma_i'}$ is the restriction operator from Γ to Γ_i'.

4 FETI-DP with corners, average edges and faces constraints

We now design a FETI-DP method for solving (4). We follow to the abstract approach described in pages 160-167 in [3].

Let us define the set of indices $\mathscr{E}_i^0$ of pairs (j,k) of Ω_j and Ω_k, $j \neq k$, for which $\bar{E}_{ijk} := \partial F_{ij} \cap \partial F_{ik}$, for $j,k \in \mathscr{F}_i^0$, is an edge of $\partial\Omega_i$. In spite of the common edges E_{ijk}, E_{jik}, and E_{kij} being geometrically the same, we treat them separately since we consider different triangulations on $E_{ijk} \subset \partial\Omega_i$ with a mesh parameter h_i, $E_{jik} \subset \partial\Omega_j$ with a mesh parameter h_j and $E_{kij} \subset \partial\Omega_k$ with a mesh parameter h_k. We denote the interior edge nodes of these triangulations by E_{ijkh}, E_{jikh} and E_{kijh}, respectively.

Let us introduce the nodal points associated to the corner unknowns by

$$\mathscr{V}_i := \{\cup_{(j,k)\in\mathscr{E}_i^0} \partial E_{ijk}\} \text{ and } \mathscr{V}_i' := \{\mathscr{V}_i \bigcup \{\cup_{(j,k)\in\mathscr{E}_i^0} \partial E_{jik} \cup \partial E_{kij}\}\}.$$

We say that $u = \{u_i\}_{i=1}^N \in W(\Omega')$ is continuous at the corners $\mathscr{V}_i$ if

$$(u_i)_i(x) = (u_j)_i(x) = (u_k)_i(x) \quad \text{at all } x \in \mathscr{V}_i.$$

Definition 1. (Subspaces $\tilde{W}(\Omega')$ and $\tilde{W}(\Gamma')$). The $\tilde{W}(\Omega')$ consists of functions $u = \{u_i\}_{i=1}^N \in W(\Omega')$ for which, for all $1 \le i \le N$, the following conditions are satisfied:

- At all corners $\mathscr{V}_i$, u is continuous.
- On all edges E_{ijk} for $(j,k) \in \mathscr{E}_i^0$

$$(\bar{u_i})_{i,E_{ijk}} = (\bar{u_j})_{i,E_{ijk}} = (\bar{u_k})_{i,E_{ijk}}.$$

- On all faces F_{ij} for $j \in \mathscr{F}_i^0$

$$(\bar{u_i})_{i,F_{ij}} = (\bar{u_j})_{i,F_{ij}},$$

where

$$(\bar{u_i})_{i,E_{ijk}} = \frac{1}{|E_{ijk}|}\int_{E_{ijk}} (u_i)_i ds, \quad (\bar{u_j})_{i,F_{ij}} = \frac{1}{|F_{ij}|}\int_{F_{ij}} (u_j)_i ds.$$

The $\tilde{W}(\Gamma')$ denotes the subspace of $\tilde{W}(\Omega')$ of functions which are discrete harmonic in the sense of $\mathscr{H}_i'$. It is easy to see that $\hat{W}(\Gamma') \subset \tilde{W}(\Gamma') \subset W(\Gamma')$.

Let $\tilde{A}$ be the stiffness matrix which is obtained by assembling the matrices A_i' for $1 \le i \le N$, from $W(\Omega')$ to $\tilde{W}(\Omega')$. We represent $u \in \tilde{W}(\Omega')$ as $u = (u_I, u_\Pi, u_\triangle)$ where the subscript I refers to the interior degrees of freedom at the nodal points on I, the Π refers to the degrees of freedom at the corners $\{\mathscr{V}_i\}_{i=1}^N$ and edges and faces averages, and the $\triangle$ refers to the remaining degrees of freedom, i.e., the nodal

values on $\{\Gamma_i'\backslash\mathscr{V}_i'\}_{i=1}^N$ with edges and faces average equal to zero. For details on $\tilde{A}$, see (4.5) in [2], and its Schur complement $\tilde{S}$ (after eliminating the I and Π degrees of freedom from $\tilde{A}$), see (4.6) in [2].

A vector $u \in \tilde{W}(\Gamma')$ can uniquely be represented by $u = (u_\Pi, u_\triangle)$, therefore, we can represent

$$\tilde{W}(\Gamma') = \hat{W}_\Pi(\Gamma') \times W_\triangle(\Gamma'),$$

where $\hat{W}_\Pi(\Gamma')$ refers to the Π-degrees of freedom of $\tilde{W}(\Gamma')$ while $W_\triangle(\Gamma')$ to the $\triangle$-degrees of freedom of $\tilde{W}(\Gamma')$. The vector space $W_\triangle(\Gamma')$ can be decomposed as

$$W_\triangle(\Gamma') = \prod_{i=1}^N W_{i,\triangle}(\Gamma_i'),$$

where the local space $W_{i,\triangle}(\Gamma_i')$ refers to the degrees of freedom associated to the nodes of $\Gamma_i'\backslash\mathscr{V}_i'$ for $1 \le i \le N$ with zero averages on F_{ij} and F_{ji}, for $i \in \mathscr{F}_i^0$, and on E_{ijk}, E_{jik} and E_{kij}, for $(j,k) \in \mathscr{E}_i^0$.

The jump operator $B_\triangle : W_\triangle(\Gamma') \to U_r$

$$B_\triangle = (B_\triangle^{(1)}, B_\triangle^{(2)}, \cdots, B_\triangle^{(N)})$$

is defined as follows. Each $B_\triangle^{(i)}$ maps $W_\triangle(\Gamma')$ to $U_{i,r}$ (jumps on edges and faces), where $v_i = B^{(i)} u_\triangle$ is defined by:

- For each face F_{ij} for $j \in \mathscr{F}_i^0$, let

$$v_i(x) = (u_{i,\triangle})_i(x) - (u_{j,\triangle})_i(x) \quad \text{for all } x \in F_{ijh}.$$

- For each edge E_{ijk} for $(j,k) \in \mathscr{E}_i^0$, let $v_i = \{v_{i,1}, v_{i,2}\}$, where

$$v_{i,1}(x) = (u_{i,\triangle})_i(x) - (u_{j,\triangle})_i(x) \quad \text{for all } x \in E_{ijkh},$$

$$v_{i,2}(x) = (u_{i,\triangle})_i(x) - (u_{k,\triangle})_i(x) \quad \text{for all } x \in E_{ijkh}.$$

Let $U_r = (U_{1,r}, \cdots, U_{N,r})$ where $U_{i,r}$ is the range of $B_\triangle^{(i)}$, and note that the $U_{i,r}$ also has zero average on edges and faces. The space U_r will also be denoted as the space of Lagrange multipliers. We note that by setting $B_\triangle^{(i)} u_\triangle = 0$, we have one constraint for each node on F_{ijh} and two constraints for each node on E_{ijkh}. The saddle point problem is defined as in [2], except that here we replace $\hat{W}_\triangle$ by U_r, and the problem (4) is reduced to: *Find* $u_\triangle^* \in W_\triangle(\Gamma')$ *and* $\lambda^* \in U_r$ *such that*

$$\begin{cases} \tilde{S} u_\triangle^* & + B_\triangle^T \lambda^* = \tilde{g}_\triangle \\ B_\triangle u_\triangle^* & = 0. \end{cases}$$

Hence, it reduces to

$$F\lambda^* = g, \tag{5}$$

where

$$F := B_\triangle \tilde{S}^{-1} B_\triangle^T, \qquad g := B_\triangle \tilde{S}^{-1} \tilde{g}_\triangle.$$

4.1 Dirichlet Preconditioner

We now define the FETI-DP preconditioner for F, see (5). Let $S'_{i,\triangle}$ be the Schur complement of S'_i restricted to $W_{i,\triangle}(\Gamma'_i) \subset W_i(\Gamma'_i)$, and define $S'_\triangle = \mathrm{diag}\{S'_{i,\triangle}\}_{i=1}^N$.

Let us introduce diagonal scaling matrices $D_i : U_{i,r} \to U_{i,r}$, for $1 \le i \le N$ as follows. For $\beta \in [1/2, \infty)$, define the diagonal entry of D_i by:

- For each face F_{ij} for $j \in \mathscr{F}_i^0$, let

$$D_i(x) = \rho_j^\beta (\rho_i^\beta + \rho_j^\beta)^{-1} =: \gamma_{ji} \quad \text{for all } x \in F_{ijh}.$$

- For each edge E_{ijk} for $(j,k) \in \mathscr{E}_i^0$, let $D_i = \{D_{i,1}, D_{i,2}\}$, where

$$D_{i,1}(x) = \rho_j^\beta (\rho_i^\beta + \rho_j^\beta + \rho_k^\beta)^{-1} =: \gamma_{jik} \quad \text{for all } x \in E_{ijkh},$$

$$D_{i,2}(x) = \rho_k^\beta (\rho_i^\beta + \rho_j^\beta + \rho_k^\beta)^{-1} =: \gamma_{kij} \quad \text{for all } x \in E_{ijkh}.$$

We now introduce $B_{D,\triangle} : U_r \to U_r$ by $B_{D,\triangle} = (D_1 B_\triangle^{(1)}, \cdots, D_N B_\triangle^{(N)})$ and the operator $P_\triangle : W_\triangle(\Gamma') \to W_\triangle(\Gamma')$ by $P_\triangle := B_{D,\triangle}^T B_\triangle$. We can check that for $u_\triangle = \{u_{i,\triangle}\}_{i=1}^N \in W_\triangle(\Gamma')$, that $v_\triangle := P_\triangle u_\triangle$ satisfies:

$$(v_{i,\triangle})_i = \gamma_{ji}[(u_{i,\triangle})_i - (u_{j,\triangle})_i] \quad \text{on} \quad F_{ijh}, \tag{6}$$

$$(v_{j,\triangle})_i = \gamma_{ij}[(u_{j,\triangle})_i - (u_{i,\triangle})_i] \quad \text{on} \quad F_{ijh}, \tag{7}$$

$$(v_{i,\triangle})_i = \gamma_{jik}[(u_{i,\triangle})_i - (u_{j,\triangle})_i] + \gamma_{kij}[(u_{i,\triangle})_i - (u_{k,\triangle})_i] \quad \text{on} \quad E_{ijkh}, \tag{8}$$

$$(v_{j,\triangle})_i = \gamma_{ijk}[(u_{j,\triangle})_i - (u_{i,\triangle})_i] + \gamma_{kij}[(u_{j,\triangle})_i - (u_{k,\triangle})_i] \quad \text{on} \quad E_{ijkh}, \tag{9}$$

$$(v_{k,\triangle})_i = \gamma_{ijk}[(u_{k,\triangle})_i - (u_{i,\triangle})_i] + \gamma_{jik}[(u_{k,\triangle})_i - (u_{j,\triangle})_i] \quad \text{on} \quad E_{ijkh}. \tag{10}$$

We note from [(6) - (7)] that on F_{ijh} it holds

$$[(v_{i,\triangle})_i - (v_{j,\triangle})_i] = [(u_{i,\triangle})_i - (u_{j,\triangle})_i],$$

and from [(8) - (9)] + [(8) - (10)] that on E_{ijkh} it holds

$$[(v_{i,\triangle})_i - (v_{j,\triangle})_i] + [(v_{i,\triangle})_i - (v_{k,\triangle})_i] = [(u_{i,\triangle})_i - (u_{j,\triangle})_i] + [(u_{i,\triangle})_i - (u_{k,\triangle})_i],$$

and it follows that $B_\triangle P_\triangle = B_\triangle$ and $P_\triangle^2 = P_\triangle$.

In the FETI-DP method, the preconditioner M^{-1} is defined as follows:

$$M^{-1} = B_D S'_\triangle B_D^T = \sum_{i=1}^{N} D_i B_\triangle^{(i)} S'_{i,\triangle} (B_\triangle^{(i)})^T D_i.$$

The main result of this paper is the following:

Theorem 1. *For any $\lambda \in U_r$, it holds that*

$$\langle M\lambda, \lambda \rangle \ \leq \ \langle F\lambda, \lambda \rangle \ \leq \ C(1+\log \frac{H}{h})^2 \langle M\lambda, \lambda \rangle,$$

where C is a positive constant independent of h_i, H_i, λ and the jumps of ρ_i. Here and below, $\log(\frac{H}{h}) := \max_{i=1}^{N} \log(\frac{H_i}{h_i})$.

Proof. Using the same algebraic arguments as in [2], it reduces to Lemma 1. The proof of Lemma 1 for the 3-D case is new and given with details below.

Lemma 1. *For any $u_\triangle \in W_\triangle(\Gamma')$, it holds that*

$$\|P_\triangle u_\triangle\|^2_{S'_\triangle} \leq C(1+\log \frac{H}{h})^2 \|u_\triangle\|^2_{\tilde{S}}, \tag{11}$$

where C is a positive constant independent of h_i, H_i, $u_\triangle$ and the jumps of ρ_i.

Proof. Given $u_\triangle \in W_\triangle(\Gamma')$, let $u = (u_\Pi, u_\triangle) \in \tilde{W}(\Gamma')$ be the solution of

$$\langle \tilde{S} u_\triangle, u_\triangle \rangle = \min \langle S'w, w \rangle =: \langle S'u, u \rangle, \tag{12}$$

where the minimum is taken over $w = (w_\Pi, w_\triangle) \in \tilde{W}(\Gamma')$ such that $w_\Pi \in \hat{W}_\Pi(\Gamma')$ and $w_\triangle = u_\triangle$. Hence, we can replace $\|u_\triangle\|_{\tilde{S}}$ in (11) by $\|u\|_{S'}$.

Let us represent the u defined above as $\{u_i\}_{i=1}^N \in W(\Gamma')$ where $u_i \in W_i(\Gamma'_i)$. Let $v \in \tilde{W}(\Gamma')$ be equal to $P_\triangle u_\triangle$ at the $\triangle$-nodes and equal to zero at the Π-nodes, i.e., $v = 0$ on $\mathcal{V}'_i$ for $1 \leq i \leq N$ and zero average on faces and edges. Let us represent v as $\{v_i\}_{i=1}^N \in W(\Gamma')$, where $v_i \in W_i(\Gamma'_i)$. We have

$$\|P_\triangle u_\triangle\|^2_{S'_\triangle} = \|v\|^2_{S'} = \sum_{i=1}^{N} \|v_i\|^2_{S'_i}$$

in view of the definition of $S'_{i,\triangle}$ and $S_\triangle$, see (4.18), (3.5) and (4.6) in [2]. Hence, to prove the lemma it remains to show that

$$\sum_{i=1}^{N} \|v_i\|^2_{S'_i} \leq C(1+\log \frac{H}{h})^2 \|u\|^2_{S'}$$

since by (12) we obtain (11). By Corollary 3.2 in [2] we need to show

$$\sum_{i=1}^{N} \tilde{d}_i(v_i, v_i) \leq C(1+\log \frac{H}{h})^2 \sum_{i=1}^{N} \tilde{d}_i(u_i, u_i),$$

where, see (2.9) in [2], $\tilde{d}_i(v_i,v_i) = d_i(\mathscr{H}_i v_i, \mathscr{H}_i v_i)$ and

$$\tilde{d}_i(v_i,v_i) = \rho_i \parallel \nabla(\mathscr{H}_i v_i)_i \parallel^2_{L^2(\Omega_i)} + \sum_{j\in\mathscr{F}_i} \frac{\rho_i \delta}{l_{ij} h_{ij}} \parallel (v_i)_i - (v_i)_j \parallel^2_{L^2(F_{ij})} . \tag{13}$$

Here, $(v_i)_i = (\mathscr{H}_i v_i)_i$ and $(u_i)_i = (\mathscr{H}_i u_i)_i$ inside of the subdomains Ω_i.

To estimate the terms of the right-hand side (RHS) of (13) we represent $(v_i)_i$ as

$$(v_i)_i = \sum_{F_{ij}\subset(\partial\Omega_i\setminus\partial\Omega)} \theta_{F_{ij}}(v_i)_i + \sum_{E_{ijk}\subset\partial\Omega_i} \theta_{E_{ijk}}(v_i)_i \tag{14}$$

and $\mathscr{H}_i$ is discrete harmonic on Ω_i. Here, $\theta_{F_{ij}}(v_i)_i := I^{h_i}(\vartheta_{F_{ij}}(v_i)_i)$ and $\theta_{E_{ijk}}(v_i)_i := I^{h_i}(\vartheta_{E_{ijk}}(v_i)_i)$, where $\vartheta_{F_{ij}}$ and $\vartheta_{E_{ijk}}$ are the standard face and edge cutoff functions and I^{h_i} the finite element interpolator. We note that we do not have any vertex terms in (14) since $(v_i)_i = 0$ on $\mathscr{V}_i$. From now on, we denote $\nabla(\mathscr{H}_\ell w_\ell)_\ell$ by $\nabla(w_\ell)_\ell$ for $\ell = i,j,k$ and $w = v,u$. Hence, using (14), we have

$$\|\nabla(v_i)_i\|^2_{L^2(\Omega_i)} \le C\{ \sum_{j\in\mathscr{F}_i^0} \|\theta_{F_{ij}}(v_i)_i\|^2_{H^{1/2}_{00}(F_{ij})} + \sum_{(j,k)\in\mathscr{E}_i^0} \|\theta_{E_{ijk}}(v_i)_i\|^2_{L^2(E_{ijk})}\} \tag{15}$$

by well-known estimates, see [3]. Note that (15) is also valid for substructures Ω_i which intersect $\partial\Omega$ by using the same arguments as for the 2-D case; see [2]. Using (6), $(\overline{u_i})_{i,F_{ij}} = (\overline{u}_j)_{i,F_{ij}}$ and Lemma 4.26 in [3], we obtain

$$\begin{aligned} &\rho_i \parallel \theta_{F_{ij}}(v_i)_i \parallel^2_{H^{1/2}_{00}(F_{ij})} = \rho_i \gamma_{ji}^2 \parallel \theta_{F_{ij}}[(u_i)_i - (u_j)_i] \parallel^2_{H^{1/2}_{00}(F_{ij})} \\ &\le C\rho_i\gamma_{ji}^2 (1+\log\frac{H_i}{h_i})^2 |(u_i)_i - (u_j)_i|^2_{H^{1/2}(F_{ij})}. \end{aligned} \tag{16}$$

Let $Q_{i,F_{ij}}$ be the L^2-projection onto $X_i(F_{ij})$, the restriction of $X_i(\Omega_i)$ on $\overline{F}_{ij}$. Using the triangle and inverse inequalities, and the $H^{1/2}$- and L^2-stability of the $Q_{i,F_{ij}}$ projection, we have

$$\begin{aligned} &|(u_i)_i - (u_j)_i|^2_{H^{1/2}(F_{ij})} \\ &\le C\{|Q_{i,F_{ij}}[(u_i)_i - (u_j)_j]|^2_{H^{1/2}(F_{ij})} + |Q_{i,F_{ij}}[(u_j)_j - (u_j)_i]|^2_{H^{1/2}(F_{ij})} \\ &\le C\,\{|(u_i)_i|^2_{H^1(\Omega_i)} + |(u_j)_j|^2_{H^1(\Omega_j)} + \frac{1}{h_i}\|(u_j)_j - (u_j)_i\|^2_{L^2(F_{ij})}\}. \end{aligned} \tag{17}$$

Substituting (17) into (16) and using $\rho_i\gamma_{ji}^2 \le \min\{\rho_i,\rho_j\}$ if $\beta \in [1/2,\infty)$, we obtain

$$\begin{aligned} &\rho_i \|\theta_{F_{ij}}(v_i)_i\|^2_{H^{1/2}_{00}(F_{ij})} \le \\ &\le C(1+\log\frac{H_i}{h_i})^2 \quad \{\rho_i |(u_i)_i|^2_{H^1(\Omega_i)} + \rho_j |(u_j)_j|^2_{H^1(\Omega_j)} + \frac{\rho_j}{h_i}\|(u_j)_j - (u_j)_i\|^2_{L^2(F_{ij})}\} \end{aligned} \tag{18}$$

$$\leq C(1+\log\frac{H_i}{h_i})^2\{\tilde{d}_i(u_i,u_i)+\tilde{d}_j(u_j,u_j)\}.$$

We now estimate the second term of (15). Using (8), we have

$$\rho_i\|\theta_{E_{ijk}}(v_i)_i\|^2_{L^2(E_{ijk})}\leq 2\rho_i\{\gamma^2_{jik}\|(u_i)_i-(u_j)_i\|^2_{L^2(E_{ijk})}+\gamma^2_{kij}\|(u_i)_i-(u_k)_i\|^2_{L^2(E_{ijk})}\}.$$

Using $(\overline{u_i})_{i,E_{ijk}}=(\overline{u_j})_{i,E_{ijk}}$ and Lemma 4.17 in [3], and the same arguments given in (17), and $\rho_i\gamma^2_{jik}\leq\min\{\rho_i,\rho_j\}$ for $\beta\in[1/2,\infty)$, we obtain

$$\rho_i\gamma^2_{jik}\|(u_i)_i-(u_j)_i\|^2_{L^2(E_{ijk})}\leq C(1+\log\frac{H_i}{h_i})\rho_i\gamma^2_{jik}|(u_i)_i-(u_j)_i|^2_{H^{1/2}(F_{ij})} \tag{19}$$

$$\leq C(1+\log\frac{H_i}{h_i})\{\rho_i|(u_i)_i|^2_{H^1(\Omega_i)}+\rho_j|(u_j)_j|^2_{H^1(\Omega_i)}+\frac{\rho_j}{h_i}\|(u_j)_j-(u_j)_i\|^2_{L^2(F_{ij})}\}$$

$$\leq C(1+\log\frac{H_i}{h_i})\{\tilde{d}_i(u_i,u_i)+\tilde{d}_j(u_j,u_j)\}$$

and similarly

$$\rho_i\gamma^2_{kij}\|(u_i)_i-(u_k)_i\|^2_{L^2(E_{ijk})}\leq C(1+\log\frac{H_i}{h_i})\{\tilde{d}_i(u_i,u_i)+\tilde{d}_k(u_k,u_k)\}. \tag{20}$$

Hence, by adding (19) and (20), we obtain

$$\rho_i\|\theta_{E_{ijk}}(v_i)_i\|^2_{L^2(E_{ijk})}\leq C(1+\log\frac{H_i}{h_i})\{\tilde{d}_i(u_i,u_i)+\tilde{d}_j(u_j,u_j)+\tilde{d}_k(u_k,u_k)\}. \tag{21}$$

Substituting (18) and (21) into (15), we get

$$\rho_i\|\nabla(v_i)_i\|^2_{L^2(\Omega_i)}\leq C(1+\log\frac{H_i}{h_i})^2\{\tilde{d}_i(u_i,u_i)+\tilde{d}_j(u_j,u_j)+\tilde{d}_k(u_k,u_k)\}. \tag{22}$$

We now estimate the second term of the RHS of (13). Note that $(v_i)_i$ and $(v_i)_j$ are defined on different meshes. In addition, the nodal values of $(v_i)_i(x)$, are defined by different formulas if a node x belongs to F_{ijh} or to $E_{ijkh}\subset\partial F_{ij}$, see (6) and (8). The same holds for $(v_i)_j(x)$. These issues must be taken into account when estimating the second terms of the RHS of (13). We have

$$\begin{aligned}\|(v_i)_i-(v_i)_j\|^2_{L^2(F_{ij})}&\leq 2\{\|(v_i)_i-Q_{i,F_{ij}}(v_i)_j\|^2_{L^2(F_{ij})}+\|(v_i)_j-Q_{i,F_{ij}}(v_i)_j\|^2_{L^2(F_{ij})}\}\\&\equiv 2\{I+II\}.\end{aligned} \tag{23}$$

Using (14) and that $w_i=\theta_{F_{ij}}w_i+\theta_{\partial F_{ij}}w_i$ for $w_i\in X_i(\Omega_i)_{|\overline{F_{ij}}}$, we have

$$\begin{aligned}I&\leq C\{\|\theta_{F_{ij}}[(v_i)_i-Q_{i,F_{ij}}(v_i)_j]\|^2_{L^2(F_{ij})}+\|\theta_{\partial F_{ij}}[(v_i)_i-Q_{i,F_{ij}}(v_i)_j]\|^2_{L^2(F_{ij})}\}\\&\equiv C\{I_{F_{ij}}+I_{\partial F_{ij}}\}.\end{aligned} \tag{24}$$

To estimate $I_{F_{ij}}$, we first represent $(v_i)_j=\theta_{F_{ji}}(v_i)_j+\theta_{\partial F_{ji}}(v_i)_j$ to obtain

$$I_{F_{ij}} \leq 2\{\|\theta_{F_{ij}}\{(v_i)_i - Q_{i,F_{ij}}\theta_{F_{ji}}(v_i)_j\}\|^2_{L^2(F_{ij})} + \|\theta_{F_{ij}}Q_{i,F_{ij}}\theta_{\partial F_{ji}}(v_i)_j)\|^2_{L^2(F_{ij})}\}$$
$$\equiv 2\{I^{(1)}_{Fij} + I^{(2)}_{Fij}\}. \tag{25}$$

Using (6) and (7), we have

$$I^{(1)}_{Fij} \leq C\gamma^2_{ji}\|\theta_{F_{ij}}\{[(u_i)_i - (u_j)_i] - Q_{i,F_{ij}}\theta_{F_{ji}}[(u_i)_j - (u_j)_j]\}\|^2_{L^2(F_{ij})}$$

and by adding and subtracting $\theta_{F_{ij}}Q_{i,F_{ij}}\theta_{\partial F_{ji}}[(u_i)_j - (u_j)_j]$, we obtain

$$\begin{aligned}
I^{(1)}_{Fij} &\leq C\gamma^2_{ji}\{\|\theta_{F_{ij}}\{[(u_i)_i - (u_j)_i] - Q_{i,F_{ij}}[(u_i)_j - (u_j)_j])\}\|^2_{L^2(F_{ij})} + \\
&+ \|\theta_{F_{ij}}Q_{i,F_{ij}}\theta_{\partial F_{ji}}[(u_i)_j - (u_j)_j])\}\|^2_{L^2(F_{ij})}\} \\
&\leq C\gamma^2_{ji}\{\|(u_i)_i - Q_{i,F_{ij}}(u_i)_j\|^2_{L^2(F_{ij})} + \|(u_j)_i - Q_{i,F_{ij}}(u_j)_j\|^2_{L^2(F_{ij})} + \\
&+ \sum_{E_{jik}\subset\partial F_{ji}} h_j\|(u_i)_j - (u_j)_j\|^2_{L^2(E_{jik})}\} \leq C\gamma^2_{ji}\{\|(u_i)_i - (u_i)_j\|^2_{L^2(F_{ij})} + \\
&+ \|(u_j)_i - (u_j)_j\|^2_{L^2(F_{ij})} + h_j(1+\log\frac{H_j}{h_j})|(u_i)_j - (u_j)_j|^2_{H^{1/2}(F_{ji})}\} \\
&\leq C\gamma^2_{ji}\{\|(u_i)_i - (u_i)_j\|^2_{L^2(F_{ij})} + \|(u_j)_i - (u_j)_j\|^2_{L^2(F_{ij})} + \\
&+ (1+\log\frac{H_j}{h_j})(h_j|(u_i)_i|^2_{H^1(\Omega_i)} + h_j|(u_j)_j|^2_{H^1(\Omega_j)} + \|(u_i)_i - (u_i)_j\|^2_{L^2(F_{ij})}\},
\end{aligned} \tag{26}$$

where we have used the L^2-stability of $Q_{i,F_{ij}}$ and $\theta_{F_{ji}}$, the constraint $(\overline{u_i})_{j,E_{jik}} = (\overline{u_j})_{j,E_{jik}}$ and Lemma 4.17 in [3]. For the last inequality of (26), we have used a similar argument as in (17).

To estimate $I^{(2)}_{F_{ij}}$, first note that

$$I^{(2)}_{F_{ij}} \leq Ch_j\|(v_i)_j\|^2_{L^2(\partial F_{ji})} \leq Ch_j\sum_{E_{jik}\subset\partial F_{ji}}\|(v_i)_j\|^2_{L^2(E_{jik})} \tag{27}$$

and using the definition of $(v_j)_i$, see (9), we have

$$\|(v_i)_j\|^2_{L^2(E_{jik})} \leq 2\{\gamma^2_{jik}\|(u_i)_j - (u_j)_j\|^2_{L^2(E_{jik})} + \gamma^2_{kij}\|(u_i)_j - (u_k)_j\|^2_{L^2(E_{jik})}\}. \tag{28}$$

The first term of the RHS of (28) is estimated as in (19) while the second term as

$$\begin{aligned}
h_j\|(u_i)_j - (u_k)_j\|^2_{L^2(E_{jik})} &\leq 2h_j\{\|(u_i)_j - (u_j)_j\|^2_{L^2(E_{jik})} + \|(u_j)_j - (u_k)_j\|^2_{L^2(E_{jik})}\} \\
\leq C(1+\log\frac{H_j}{h_j})\{h_j|(u_i)_i|^2_{H^1(\Omega_i)} + h_j|(u_j)_j|^2_{H^1(\Omega_j)} + \|(u_i)_j - (u_i)_i\|^2_{L^2(F_{ji})} \\
+h_j|(u_k)_k|^2_{H^1(\Omega_j)} + \|(u_k)_j - (u_k)_k\|^2_{L^2(F_{jk})}\}.
\end{aligned} \tag{29}$$

Substituting (28) and (29) into (27) and adding with (26), see (25), we obtain

$$\frac{\rho_i\delta}{l_{ij}h_{ij}}I_{F_{ij}} \le C(1+\log\frac{H}{h})\{\frac{h_j}{h_{ij}}\tilde{d}_i(u_i,u_i)+\frac{h_j}{h_{ij}}\tilde{d}_j(u_j,u_j)+\sum_{E_{ijk}\subset\partial F_{ij}}\frac{h_j}{h_{ij}}\tilde{d}_k(u_k,u_k)\}.$$

We now estimate $I_{\partial F_{ij}}$, see (24). Note that $(\overline{v_i})_{j,F_{ji}}=0$ implies a zero average of $Q_{i,F_{ij}}(v_i)_j$ on F_{ij}. We also have $(\overline{v_j})_{i,F_{ij}}=0$. Using previous arguments, we obtain

$$\begin{aligned}
&I_{\partial F_{ij}} \le Ch_i\|(v_i)_i - Q_{i,F_{ij}}(v_i)_j\|^2_{L^2(\partial F_{ij})}\\
&\le Ch_i\{\|(v_i)_i\|^2_{L^2(\partial F_{ij})}+\|Q_{i,F_{ij}}\theta_{F_{ji}}(v_i)_j\|^2_{L^2(\partial F_{ij})}+\|Q_{i,F_{ij}}\theta_{\partial F_{ji}}(v_i)_j\|^2_{L^2(\partial F_{ij})}\}\\
&\le C\sum_{E_{ijk}\subset\partial F_{ij}}\{h_i\|(v_i)_i\|^2_{L^2(E_{ijk})}+h_i\|Q_{i,F_{ij}}\theta_{F_{ji}}(v_i)_j\|^2_{L^2(E_{ijk})}+h_j\|(v_i)_j\|^2_{L^2(E_{jik})}\}\\
&\equiv C\sum_{E_{ijk}\subset\partial F_{ij}}\{I^{(1)}_{E_{ijk}}+I^{(2)}_{E_{ijk}}+I^{(3)}_{E_{ijk}}\}. \qquad (30)
\end{aligned}$$

It is not hard to see, using the same argument as priviously, that

$$\begin{aligned}
&I^{(1)}_{E_{ijk}} = h_i\|\gamma_{jik}[(u_i)_i-(u_j)_i]+\gamma_{kij}[(u_i)_i-(u_k)_i]\|^2_{L^2(E_{ijk})} \qquad (31)\\
&\le C(1+\log\frac{H_i}{h_i})\{\gamma^2_{jik}(h_i|(u_i)_i|^2_{H^1(\Omega_i)}+h_i|(u_j)_j|^2_{H^1(\Omega_j)}+\|(u_j)_j-(u_j)_i\|^2_{L^2(F_{ij})})\\
&+\ \gamma^2_{kij}(h_i|(u_i)_i|^2_{H^1(\Omega_i)}+h_i|(u_k)_k|^2_{H^1(\Omega_k)}+\|(u_k)_k-(u_k)_i\|^2_{L^2(F_{ik})})\},
\end{aligned}$$

$$\begin{aligned}
&I^{(2)}_{E_{ijk}} \le Ch_i\gamma^2_{ji}\|Q_{i,F_{ij}}\theta_{F_{ji}}[(u_j)_j-(u_i)_j]\|^2_{L^2(E_{ijk})} \qquad (32)\\
&\le Ch_i\gamma^2_{ji}\{\|Q_{i,F_{ij}}[(u_j)_j-(u_i)_j]\|^2_{L^2(E_{ijk})}+\|Q_{i,F_{ij}}\theta_{\partial_{F_{ji}}}[(u_j)_j-(u_i)_j]\|^2_{L^2(E_{ijk})}\}\\
&\le C\gamma^2_{ji}\{h_i(1+\log\frac{H_i}{h_i})|(u_j)_j-(u_i)_j\|^2_{H^{1/2}(F_{ji})}+h_j\|(u_j)_j-(u_i)_j\|^2_{L^2(E_{ijk})}\}\\
&\le C\gamma^2_{ji}(h_i+h_j)(1+\log\frac{H}{h})|(u_j)_j-(u_i)_j|^2_{H^{1/2}(F_{ji})}\\
&\le C\gamma^2_{ji}(h_i+h_j)(1+\log\frac{H}{h})\{|(u_i)_i|^2_{H^1(\Omega_i)}+|(u_j)_j|^2_{H^1(\Omega_j)}+\frac{1}{h_i}\|(u_i)_i-(u_i)_j\|^2_{L^2(F_{ij})}\},
\end{aligned}$$

$$\begin{aligned}
I^{(3)}_{E_{ijk}} &\le Ch_j\|(\gamma_{jik}[(u_i)_j-(u_j)_j]+\gamma_{kij}[(u_i)_j-(u_k)_j])\|^2_{L^2(E_{jik})} \le C(1+\log\frac{H_j}{h_j})*\\
&\{(\gamma^2_{jik}+\gamma^2_{jik})(h_j|(u_i)_i|^2_{H^1(\Omega_i)}+h_j|(u_j)_j|^2_{H^1(\Omega_j)}+\|(u_i)_i-(u_i)_j\|^2_{L^2(F_{ij})})\\
&+\gamma^2_{kij}(h_j|(u_i)_i|^2_{H^1(\Omega_i)}+h_j|(u_k)_k|^2_{H^1(\Omega_k)}+\|(u_k)_k-(u_k)_j\|^2_{L^2(F_{jk})})\}. \qquad (33)
\end{aligned}$$

Substituting (31), (32) and (33) into (30), we obtain

$$\frac{\rho_i\delta}{l_{ij}h_{ij}}I_{\partial F_{ij}} \le C(1+\log\frac{H}{h})\{\frac{h_i+h_i}{h_{ij}}(\tilde{d}_i(u_i,u_i)+\tilde{d}_j(u_j,u_j))+\sum_{E_{ijk}\subset\partial F_{ij}}\frac{h_{jk}}{h_{ij}}\tilde{d}_k(u_k,u_k)\}.$$

It remains to estimate II in (23). Using a L^2-projection property, we have

$$\begin{aligned} II &\leq Ch_i|(v_i)_j|^2_{H^{1/2}(F_{ji})} \leq C\{h_i|\theta_{F_{ji}}(v_i)_j|^2_{H^{1/2}(F_{ji})} + h_i|\theta_{\partial F_{ji}}(v_i)_j|^2_{H^{1/2}(F_{ji})}\} \\ &\equiv C\{II_{F_{ji}} + II_{\partial F_{ji}}\}. \end{aligned} \tag{34}$$

Using similar arguments as above, we obtain

$$II_{F_{ji}} \leq Ch_i(1+\log\frac{H_j}{h_j})^2\gamma_{ji}^2|(u_i)_j-(u_j)_j|^2_{H^{1/2}(F_{ji})} \tag{35}$$

$$\leq C(1+\log\frac{H_j}{h_j})^2\gamma_{ji}^2\{h_i|(u_i)_i|^2_{H^1(\Omega_i)} + h_i|(u_j)_j|^2_{H^1(\Omega_j)} + \frac{h_i}{h_j}\|(u_i)_i-(u_i)_j\|^2_{L^2(F_{ij})}\},$$

$$II_{\partial F_{ji}} \leq C\frac{h_i}{h_j}\|\theta_{\partial F_{ji}}(v_i)_j\|^2_{L^2(F_{ji})} \leq Ch_i\sum_{E_{jik}\subset\partial F_{ji}}\|(v_i)_j\|^2_{L^2(E_{jik})}, \tag{36}$$

and

$$\begin{aligned} h_i\|(v_i)_j\|^2_{L^2(E_{jik})} &\leq C(1+\log\frac{H_j}{h_j})\{(\gamma_{jik}+\gamma_{kij})* \\ &(h_i|(u_i)_i|^2_{H^1(\Omega_i)} + h_i|(u_j)_j|^2_{H^1(\Omega_j)} + \frac{h_i}{h_j}\|(u_i)_j-(u_i)_i\|^2_{L^2(F_{ji})}) \\ &+\gamma_{kij}(h_i|(u_k)_k|^2_{H^1(\Omega_k)} + h_i|(u_j)_j|^2_{H^1(\Omega_j)} + \frac{h_i}{h_j}\|(u_k)_j-(u_k)_k\|^2_{L^2(F_{jk})})\}. \end{aligned} \tag{37}$$

Substituting (37) into (36) and adding (35), see (34), we obtain

$$\frac{\rho_i\delta}{l_{ij}h_{ij}}II \leq C(1+\log\frac{H_j}{h_j})(\frac{h_i}{h_{ij}}\tilde{d}_i(u_i,u_i) + \frac{h_i}{h_{ij}}\tilde{d}_j(u_j,u_j) + \sum_{E_{ijk}\subset\partial F_{ij}}\frac{h_{jk}}{h_{ij}}\frac{h_i}{h_j}\tilde{d}_k(u_k,u_k)\}.$$

The proof is complete.

Remark 1. The proof of Lemma 1 also works with minor modifications when $\bar{F}_{ij} = \partial\Omega_i \cap \partial\Omega_j$ is an union of faces, also, for FETI-DP with corner and average face constraints only, or with corner and edge constraints only.

Acknowledgements The first author has been partially supported by the Polish NSC grant 2011/01/B/ST1/01179.

References

1. Dryja, M.: On discontinuous Galerkin methods for elliptic problems with discontinuous coefficients. Comput. Methods Appl. Math. **3**(1), 76–85 (electronic) (2003). Dedicated to Raytcho Lazarov

2. Dryja, M., Galvis, J., Sarkis, M.: A FETI-DP preconditioner for a composite finite element and discontinuous Galerkin method. SIAM J. Numer. Anal. **51**(1), 400–422 (2013). DOI 10.1137/100796571. URL http://dx.doi.org/10.1137/100796571
3. Toselli, A., Widlund, O.: Domain decomposition methods—algorithms and theory, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin (2005)

A Multi-Stage Preconditioner for the Black Oil Model and Its OpenMP Implementation

Chunsheng Feng[1], Shi Shu[1], Jinchao Xu[2], and Chen-Song Zhang[3]

1 Introduction

A significant portion of our energy needs is met using oil and gas, and mathematical models of flow through porous media play an important role in developing and managing oil and gas reservoirs. Highly sophisticated mathematical and computational methods that describe compressible multi-phase multi-component fluid flow in reservoirs are crucial for optimizing oil reservoir development. Numerical solutions of these highly nonlinear coupled partial differential equations (PDEs) require moderate to sophisticated algorithms and computing platforms.

When a reservoir's pressure drops below bubble-point pressure, the hydrocarbon phase splits into a liquid (oil) phase and a gaseous (gas) phase at the thermodynamical equilibrium. Under these conditions, the flow in the porous media is of the black oil type: the water phase does not exchange mass with the other phases, and the liquid and gaseous phases exchange mass with each other. This model is referred to as the black oil model and is often applied in primary and secondary oil recovery. In this paper, we will consider a numerical solution of the black oil model, although the methods discussed here can be extended to other models.

We propose an algorithm for solving the Jacobian system $Ax = b$ arising from the fully implicit method, which is the most popular method for the black oil model (see [8]). The proposed method constructs an efficient preconditioner using the framework in [14]. We will focus on the multithread implementation of this method in modern multicore computer environments. In order to facilitate the discussion and emphasize the main points, we will use a simplified version of the algorithm.

Obtaining a solution of a large-scale reservoir simulation is challenging. The Jacobian system resulting from the Newton linearization is usually large, sparse, highly nonsymmetric, and ill-conditioned. However, the Krylov subspace methods, such as BiCGstab and GMRes, are efficient iterative methods for these linear systems (see [21]). In order to solve a linear algebraic system of equations efficiently, a preconditioner is often necessary to accelerate a Krylov subspace method. A preconditioner is an approximation to A^{-1}, and its action on a vector should be easy to compute. The preconditioners used in reservoir simulators mainly fall into two

[1]School of Mathematics and Computational Science, Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan University, China. e-mail: {spring}{shushi}@xtu.edu.cn .[2]Department of Mathematics, The Pennsylvania State University, USA. e-mail: xu@math.psu.edu .[3]NCMIS & LSEC, Academy of Mathematics and System Sciences, China. e-mail: zhangcs@lsec.cc.ac.cn

categories: (i) purely algebraic preconditioners and (ii) preconditioners based on the different properties of the variables.

Category (i) includes block incomplete lower-upper factorization (BILU) methods [17, 10], nest factorization [3, 4], and SVD-reduction methods [24]. Category (ii), on the other hand, includes methods based on the understanding that pressure variables and saturation variables differ from each other in regard to analytic properties; representative examples are the combinative method [6], the constrained pressure residual (CPR) method [23], and several multi-stage methods [2, 16, 15, 22]. As a key component of these preconditioners, algebraic multigrid (AMG) methods [7, 20, 12] have also been applied.

There is a trend toward using multicore processors, which helps CPU designers to avoid the high power-consumption problem that comes with increasing chip frequency. As CPU speeds rise into the 3–4 GHz range, the amount of electrical power required is prohibitive. Hence, the trend toward multicore processors started and will continue into the foreseeable future. OpenMP is an application program interface that can be used to explicitly direct multicore (shared memory) parallelism. It is a specification for a set of compiler directives, library routines, and environment variables that can be used to specify shared memory parallelism in Fortran and C/C++ programs.

Several difficulties can arise when using multithread implementation for preconditioned Krylov subspace methods: (i) Some preconditioners use sequential algorithms, like Gauss-Seidel; (ii) OpenMP programs sometimes require more memory space than their corresponding sequential versions do. When a numerical algorithm is implemented in OpenMP or any other multithread computer language, it is important to maintain the convergence rate of the corresponding sequential algorithm. However, this is not always possible as many numerical algorithms are sequential in nature. When working with sparse matrices in compressed formats, like the Compressed Sparse Row format, we sometimes need to introduce auxiliary memory space. This becomes an increasingly heavy burden as the number of threads increases. We will analyze the parallel interpolation and coarse-grid operators in the setup phase of AMG based on the fact that the coefficient matrices A we consider are banded. Our results will offer a basis for reducing memory costs.

The rest of the paper is organized as follows: In Section 2, we describe the widely used black oil model and its fully implicit discretization. In Section 3, we introduce a simplified version of the preconditioner studied in [14] for the black oil model and show how this method relates to a few well-known methods such as the CPR method. In Section 4, we give the implementation details of the proposed preconditioner in the shared-memory architecture using OpenMP. Finally, in Section 5, we report the results of some numerical experiments conducted in a typical multicore computing environment.

2 The black oil model

The black oil model is developed based on the assumptions that (i) the reservoir is isothermal, (ii) the flow in porous media has three phases (liquid, gaseous, water) and three components (oil, gas, water), (iii) mass transfer occurs between the oil and gas phase, and (iv) no mass transfer occurs between the water phase and either the gas or the oil phases. We use lower- and upper-case subscripts to indicate three phases—water, oil (the liquid phase), and gas (the gaseous phase)—and the component of each—water, oil, and gas, respectively.

Let ϕ and k denote the porosity and permeability, respectively, of the porous medium $\Omega \subset \mathbb{R}^3$. For the α-phase ($\alpha = w,o,g$), let S_α, μ_α, p_α, u_α, B_α, ρ_α, and $k_{r\alpha}$ be the saturation, viscosity, pressure, volumetric velocity, formation volume factor (FVF), density, and relative permeability, respectively. Moreover, we use R_{so} to denote the gas solubility, and we use Q_{Ws}, Q_{Os}, and Q_{Gs}[1] to denote the volumetric production rate of water, oil, and gas, respectively. The mass conservation equations of the black oil model can be written as follows:

$$\frac{\partial}{\partial t}\left(\phi \frac{S_w}{B_w}\right) + \nabla \cdot \left(\frac{1}{B_w} u_w\right) = \frac{Q_{Ws}}{B_w}, \tag{1}$$

$$\frac{\partial}{\partial t}\left(\phi \frac{S_o}{B_o}\right) + \nabla \cdot \left(\frac{1}{B_o} u_o\right) = \frac{Q_{Os}}{B_o}, \tag{2}$$

$$\frac{\partial}{\partial t}\left[\phi\left(\frac{S_g}{B_g} + \frac{R_{so}S_o}{B_o}\right)\right] + \nabla \cdot \left(\frac{1}{B_g} u_g + \frac{R_{so}}{B_o} u_o\right) = \frac{Q_{Gs}}{B_g} + \frac{R_{so}Q_{Os}}{B_o}, \tag{3}$$

where

$$u_\alpha = -\frac{kk_{r\alpha}}{\mu_\alpha}\left(\nabla P_\alpha - \rho_\alpha \mathfrak{g} \nabla z\right), \qquad \alpha = w,o,g \tag{4}$$

$$S_w + S_o + S_g = 1. \tag{5}$$

Equations (1)–(3) describe the mass conservation of the water, oil, and gas components, respectively; (4) is the Darcy's law for porous media; and (5) represents the phase saturation balance. Throughout this paper, we assume that the capillary pressure between each phase is zero, i.e., $P_w = P_o = P_g = P$.

Among the many possible discretization methods for the above model, we consider only the Fully Implicit method (FIM) [11] in which the Newton linearization is combined with first-order upstream-weighting finite difference spatial discretization; for details, see [8, Chapter 8]. For the sake of simplicity and clarity, we make two more assumptions:

- All three phases are present during the whole simulation period of the black oil model; i.e., the transition between the two-phase and the three-phase regions is ignored.

[1] The subscript s indicates that these variables are at the standard conditions instead of reservoir conditions.

- The well flow rate constraints are modeled by the Peaceman model (see [19]), and they are treated explicitly; i.e., the well constraints do not contribute to the Jacobian system.

Remark 1 (Phase transition and implicit wells). We note that these two assumptions are made only so that we can the main ideas of the method as clearly as possible. In practical implementation, none of these assumptions is applicable: (i) When only two phases are present in a reservoir grid-cell, we add another primary variable—the gas solubility R_{so} or the bubble-point pressure P_b—besides oil pressure and saturation as many other simulators do. (ii) Treating well constraints implicitly is important to obtain accurate simulation results in a more stable fashion. When implicit well constraints are present, we get a bordered coefficient matrix; details on how to treat them can be found in [14].

We eliminate S_g from (1)–(4) using (5) and plug (4) into (1)–(3). Moreover, we choose the increments δP, δS_w, and δS_o as the main solution variables[2] and give the rest of the variables in terms of these main solution variables. In each Newton iteration, this discretization method gives a Jacobian system of the following type:

$$A = \begin{bmatrix} A_{1P} & A_{1S_w} & \\ A_{2P} & A_{2S_w} & A_{2S_o} \\ A_{3P} & A_{3S_w} & A_{3S_o} \end{bmatrix}, \tag{6}$$

where A_{1P} is the pressure block of the water mass conservation equation; the block matrix

$$\begin{bmatrix} A_{2S_w} & A_{2S_o} \\ A_{3S_w} & A_{3S_o} \end{bmatrix}$$

is the saturation block; and A_{1S_w}, A_{2P}, and A_{3P} are the blocks that couple the pressure with the non-pressure variables.

The coefficient matrix A of the Jacobian system is often large and sparse, and it is stored in the block compressed sparse row (BCSR)[3] format. From this point on, N_P is used to refer to the total number of pressure unknowns and N_{S_w} and N_{S_o} are the numbers of the water and oil saturation unknowns, respectively. We further define $N_S = N_{S_w} + N_{S_o}$ and $N = N_P + N_S$.

Remark 2 (Decoupling strategies). The decoupling technique is a preprocessing step designed to weaken the coupling between different unknowns. There are many possible options for decoupling, such as Householder transformations, the IMPES-type method, and the BSD method based on the least square method. Details regarding the performance of each and a comparison between them can be found in [2, 16], for example. For the present study, we apply the alternative block factorization (ABF) strategy introduced by Bank et al. [5] due to its simplicity and reasonable decoupling effects. Investigating efficient and robust decoupling strategies is beyond the scope of this paper.

[2] We denote the solution variable as $x := [\delta P, \delta S_w, \delta S_o]^T$.

[3] This data structure is similar to the compressed sparse row (CSR) format, but each nonzero entry is a 3×3 sub-matrix in BCSR.

3 A multi-stage preconditioner for FIM

It is natural to introduce auxiliary or fictitious problems for different physical unknowns and use them to construct a multi-stage (multiplicative) preconditioner. Assume that we have the transfer operators Π_P and Π_S from x to the pressure variable P and the saturations, respectively. Let R be a relaxation or smoother for A. A multi-stage preconditioner can be defined in Algorithm 1.

Algorithm 1: A multiplicative preconditioner for the black oil model

Step 0. Given an initial guess x

Step 1. $x \leftarrow x + \Pi_S B_S \Pi_S^T (b - Ax)$

Step 2. $x \leftarrow x + \Pi_P B_P \Pi_P^T (b - Ax)$

Step 3. $x \leftarrow x + R(b - Ax)$

It is easy to see that this algorithm defines a preconditioner B such that

$$I - BA = (I - RA)(I - \Pi_P B_P \Pi_P^T A)(I - \Pi_S B_S \Pi_S^T A). \tag{7}$$

The choice of auxiliary problems and their corresponding solvers is crucial to the overall performance of the preconditioner B. The auxiliary problems should preserve the property of the governing equations of each unknown. We expect A_{1P} to preserve the ellipticity of the pressure equation, and we expect multilevel solvers like AMG to solve this auxiliary problem efficiently.

To facilitate our discussion on OpenMP implementation in the next section, we will use a simple version of Algorithm 1, in which we define

$$\Pi_P = \begin{bmatrix} I_P \\ 0 \end{bmatrix} \in \mathbb{R}^{N \times N_P} \quad \text{and} \quad \Pi_S = \begin{bmatrix} 0 \\ I_S \end{bmatrix} \in \mathbb{R}^{N \times N_S},$$

where $I_P \in \mathbb{R}^{N_P \times N_P}$ and $I_S \in \mathbb{R}^{N_S \times N_S}$ are identity matrices corresponding to the pressure variables and the saturation variables, respectively. We use one classical AMG V-cycle [20] as the subspace solver B_P, and we apply the block Gauss-Seidel (GS) method as the subspace solver B_S and the relaxation R. For the multithreaded version, the usual GS method is replaced by the hybrid GS method.[4] This preconditioner is referred to as B_{MSP} in the rest of this paper.

Remark 3 (CPR preconditioner). One well-known special case of Algorithm 1 is the constrained pressure residual (CPR) preconditioner [23], which can be presented in the following algebraic form:

$$B_{\mathrm{CPR}} = R(I - AM) + M, \tag{8}$$

where

[4] The standard GS sweep is applied in each thread, and parallel (simultaneous) updating is used across multiple threads.

$$M = \begin{bmatrix} B_P & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{N\times N} \text{ and } B_P \approx A_{1P}^{-1} \text{ is constructed using AMG.}$$

The smoother R is usually defined by the Line SOR smoother or the Incomplete Factorization methods. B_P can often be replaced by one or more AMG cycles. If we choose $\Pi_P = [I_P, 0, 0]^T$, then we can rewrite the CPR preconditioner as

$$I - B_{\text{CPR}}A = (I - RA)\Big(I - \Pi_P B_P \Pi_P^T A\Big), \tag{9}$$

which has the exact same form of (7) as for A.

Remark 4 (Block triangular preconditioner). Another simple way to construct an efficient preconditioner is to choose $R = 0$ in (7). In this case, the resulting preconditioner B_{TRIG} can be viewed as a block upper triangular preconditioned with B_P as an approximated A_{1P}^{-1} and B_S as an approximation of $[A_{2S_w}, A_{2S_o}; A_{3S_w}, A_{3S_o}]^{-1}$. The preconditioner, therefore, is an inexact version of the block GS method.

4 Implementation details in OpenMP

In this section, we discuss an OpenMP implementation of the proposed auxiliary space preconditioner in Algorithm 1. Using a shared-memory paradigm can greatly simplify the programming task compared to message-passing implementations. OpenMP parallel programs are relatively easy to implement, as each processor has a global view of the entire memory. Parallelism can be achieved by inserting compiler directives into the code to distribute loop iterations among the processors. However, performance may suffer from the poor spatial locality of physically distributed shared data [18].

In this paper, we will not discuss general tasks such as sparse-matrix multiplications for OpenMP. Interested readers are referred to Oliker et al. [18] and references therein for related discussions. We will focus on one part of our algorithm, namely the setup stage of the classical AMG method and propose a simple but efficient algorithm for constructing standard prolongation and coarse-level operators using OpenMP. We show that if the bandwidth of the sparse coefficient matrix A is relatively small, then much less memory is needed.

Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric. Let $G_A(V,E)$ denote the graph of the matrix A where V is the set of vertices (i.e., unknowns), and let E be the set of edges (i.e., connections that correspond to nonzero matrix entries). Suppose the index set of vertices V is split into a set C of coarse-level vertices and a set F of fine-level vertices, such that

$$V = C \cup F \quad \text{and} \quad C \cap F = \varnothing,$$

and we denote n_c as the cardinality of C, i.e., the number of C-vertices. Assume that F^C is the map from F-vertices to C-vertices.

We define $N_i := \{j \in V : A_{ij} \neq 0,\ j \neq i\}$, and for $\theta \in [0,1)$ we denote

$$S_i(\theta) := \left\{ j \in N_i : -A_{ij} \geq \theta \cdot \max_{k \neq i}(-A_{ik}) \right\}.$$

Let $D_i^{F,s} := S_i(\theta) \cap F$, $D_i^{C,s} := S_i(\theta) \cap C$ and $D_i^w := N_i \setminus \left(D_i^{C,s} \cup D_i^{F,s}\right)$. We can now define

$$F_i := \left\{ j \in D_i^{F,s} : \ i \text{ and } j \text{ without the same depended } C\text{-vertices} \right\}.$$

Let $\hat{A}_{ij} := 0$ if $A_{ii}A_{ij} > 0$, and let $\hat{A}_{ij} := A_{ij}$ otherwise. We denote $P = (P_{ij_c}) \in \mathbb{R}^{n \times n_c}$ as the standard prolongation matrix where entry

$$P_{ij_c} = \begin{cases} \frac{-1}{A_{ii} + \sum\limits_{k \in D_i^w \cup F_i} A_{ik}} \left(A_{ij} + \sum\limits_{k \in D_i^{F,s} \setminus F_i} \frac{A_{ik}\hat{A}_{kj}}{\sum\limits_{m \in D_i^{C,s}} \hat{A}_{km}} \right), & i \in F,\ j \in D_i^{C,s},\ j_c = F^C[j], \\ 1.0, & i \in C,\ j_c = F^C[i], \\ 0.0, & \text{otherwise}. \end{cases}$$

As the matrix P is sparse and stored in the CSR format, we need to use an auxiliary integer marker called M_P to quickly locate the column index of each non-zero entry (see for example in BoomerAMG of hypre [1]). In fact, to generate the i-th row of P, we define that, for $0 \leq j \leq n-1$,

$$M_P[j] := \begin{cases} J_{j_c}, & j \in D_i^{C,s},\ j_c = F^C[j], \\ -2-i, & j \in D_i^{F,s} \setminus F_i, \\ -1, & \text{otherwise}, \end{cases} \tag{10}$$

where J_{j_c} is the position of P_{ij_c} entry in the column index array of the CSR storage of P. In the OpenMP implementation, we have to allocate the marker M_P for all OpenMP threads. The length of each M_P is n, and the total length of M_P for all threads is then $N_T \times n$ where N_T is the total number of OpenMP threads. When N_T is large, the memory cost for M_P is considerable.

Assume that $b_n = b_l + b_r$ is the bandwidth of A, where b_l and b_r are the left and right bandwidths for matrix A, respectively. When the parallel partition of V is continuously distributed in a balanced fashion to each OpenMP thread (i.e., the size difference between each thread does not exceed one), we can easily see that the length of M_P that is actually used is much smaller than n (Fig. 1). Taking into account that the matrix is banded, we can get the following estimates of the length L_P^t and the minimal offset $M_l^t(P)$[13]:

$$L_P^t \leq \min(n, \frac{n}{N_T} + 2b_n) \quad \text{and} \quad M_l^t(P) \geq \max\left(0, \frac{n}{N_T}(t-1) - 2b_n\right). \tag{11}$$

The coarse grid operator for the multigrid method can be built using the Galerkin relation $A_c = (A_{ij}^c)_{n_c \times n_c} := P^T AP$, where

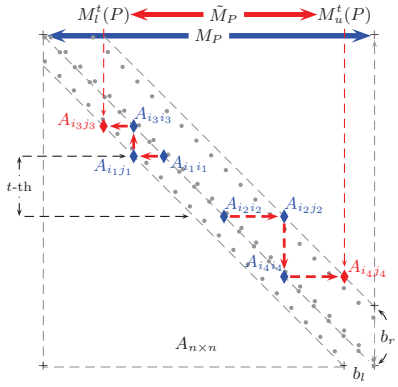


Fig. 1 Construction of the prolongation operator P for the banded sparse matrix A. Here, $M_l^t(P)$ and $M_u^t(P)$ are the lower and upper column indices, respectively, of the non-zero entries in A of the t-th OpenMP thread.

$$A_{ij}^c = \sum_{k_1}\sum_{l_1} P_{k_1 i} A_{k_1 l_1} P_{l_1 j}, \qquad i,j = 1,\cdots,n_c. \tag{12}$$

Similar to the implementation of the prolongation operator, we need to allocate two

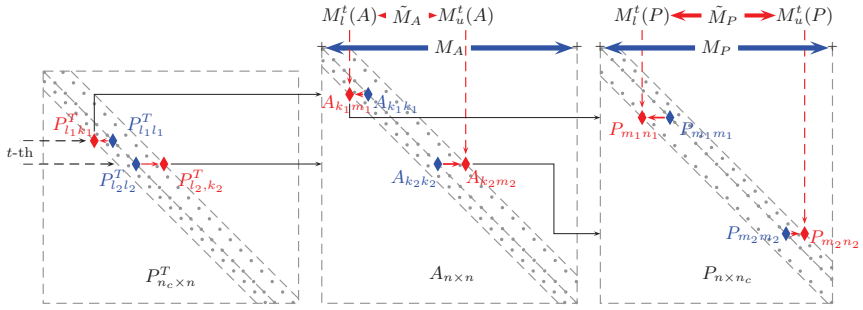


Fig. 2 Construction of the Galerkin coarse-level operator $A_c = P^T AP$. Here, $M_l^t(A)$ and $M_u^t(A)$ are the lower and upper column indices of the non-zero entries in A of the t-th OpenMP thread.

auxiliary arrays called M_A and M_P (Fig. 2). The length of M_A is n and the length of M_P is n_c. By taking into account the characteristic of the banded sparse matrices of the coarse operator, we can get the estimation formula for M_A and M_P. The actual length L_A^t and the offset $M_l^t(A)$ can be calculated using

$$L_A^t \le \min(n, \frac{n}{N_T} + 2b_n) \quad \text{and} \quad M_l^t(A) \ge \frac{n}{N_T}t - b_n. \tag{13}$$

Remark 5 (How much memory can we save?). If we do not consider the possibility that the bandwidth of A can be much smaller than n, then we will need two auxiliary arrays with length nN_T. However, as noted above, we only need two arrays of length $n + 2b_nN_T$. When $n \gg b_n$ and N_T is relatively large, we can save a lot of memory by using these improved estimates. In fact, this will reduce not only storage cost but also the time needed to allocate and initialize memory.

5 Numerical experiments

In this section, we design several numerical experiments and analyze the performance of OpenMP implementation of the preconditioner proposed in Section 3. We use a HP desktop PC equipped with two Intel Xeon X5676 (3.07GHz, 12 cores) and 96GB RAM. The experimental environment is Cent OS 6.2 and GCC 4.4.6 (with an "–O2" optimization parameter).

Our example is adapted from the second data set of the Tenth SPE Comparative Solution Project ([9]), which is designed to compare the ability of upscaling approaches used by various participants to predict the performance of water-flooding in a simple but highly heterogeneous black oil reservoir described by a fine-scale $(60 \times 220 \times 85)$ regular Cartesian geological model. This model has a simple geometry, with no top structure or faults. The model dimensions are $1200 \times 2200 \times 170$ (ft). The top 70 ft (35 layers) represents the Tarbert formation, and the bottom 100 ft (50 layers) represents the Upper Ness formation. There is one injector in the center of the field and a producer located at each of the four corners. The total simulation time is 2,000 days. The purpose of this benchmark is to compare the models in regard to accuracy and computational cost.

For our purpose, we modify the SPE10 example as a three-phase black oil test by changing the properties of the fluid. Hence, the total number of unknowns of each Jacobian system is $N = 3.3$M and the size of the pressure equation is $n = N_P = 1.1$M. We employ the GMRes method as our iterative solver for solving linear Jacobian systems. The stopping criteria is that the relative residual in the Euclidian norm is less than 10^{-4}. In Table 1, we summarize the performance of our simulator, in which #Timesteps is the total number of time steps, #Newton is the total number of Newton iterations, #Linear is the total number of linear iterations, Solver Time is the total wall-time for the linear solution steps, Aver. Newton is the average number of Newton iterations in each time step, and Aver. Linear Iter is the average number of linear iterations in each Newton iteration.

Table 1 Performance of preconditioned GMRES for solving the three-phase SPE10 problem.

Preconditioner	#Timesteps	#Newton	#Linear	Solver Time (hour)	Aver. Newton	Aver. Linear Iter
B_{MSP}	736	997	32829	6.60	1.35	32.92
B_{CPR}	796	1253	57723	20.15	1.57	41.50
B_{TRIG}	805	2045	103249	17.47	2.54	46.34

In order to further demonstrate the performance of the proposed preconditioner, we select four typical Jacobian linear systems from different periods of the 2,000 days of simulation. They are all from the first Newton iteration in different time levels and the time step sizes are the same (each is five days). Using these examples, we test the performance of the three different preconditioners, B_{MSP}, B_{CPR}, and B_{TRIG}, given in Section 3. The proposed preconditioner in Algorithm 1 results in various preconditioners depending on the different choices of auxiliary problem solvers/smoothers. In this section, we only compare the performance of these three simple choices.

The total number of iterations and the wall-time in seconds for each of these methods is reported in Tables 2–4, in which N_T is the total number of OpenMP threads. Moreover, the respective OpenMP speedup for these methods are listed along with the wall-times. We observe that these three methods are very robust for the test problems and that their OpenMP versions can deliver about three times speedup compared with the corresponding serial versions. Furthermore, the numerical tests show that each component, B_S, B_P, and R, plays a role such that dropping any of them would result in at least 20% to 30% performance lost in CPU time. And, for more difficult problems, this drop is expected to be more severe.

Table 2 Number of iterations, wall-times (seconds), and OpenMP speedups of B_{MSP}.

	1st			2nd			3nd			4nd		
N_T	#Iter	Time	Speedup	#Iter	Time	Speedup	#Iter	Time	Speedup	#Iter	Time	Speedup
1	32	31.34	—	34	32.79	—	34	32.77	—	32	31.49	—
2	32	17.72	1.77	34	18.48	1.77	34	18.46	1.78	32	17.68	1.78
4	32	13.44	2.33	34	13.19	2.49	34	13.14	2.49	32	12.60	2.50
8	33	11.02	2.84	34	11.20	2.93	34	11.18	2.93	32	10.80	2.91
12	33	10.99	2.85	34	11.27	2.91	34	10.84	3.02	32	10.77	2.92

Table 3 Number of iterations, wall-times (seconds), and OpenMP speedups of B_{CPR}.

	1st			2nd			3nd			4nd		
N_T	#Iter	Time	Speedup	#Iter	Time	Speedup	#Iter	Time	Speedup	#Iter	Time	Speedup
1	45	39.01	—	45	38.90	—	43	37.36	—	42	36.56	—
2	45	21.95	1.78	45	21.90	1.78	43	21.00	1.78	42	20.67	1.77
4	45	15.42	2.53	45	15.44	2.52	44	15.19	2.46	42	14.56	2.51
8	45	13.12	2.97	45	13.09	2.97	44	12.86	2.90	42	12.35	2.96
12	45	13.19	2.96	45	13.18	2.95	43	12.66	2.95	42	11.93	3.07

Finally, we test the memory cost for the AMG setup stage, which is crucial in constructing B_P. As discussed in Section 4, the auxiliary arrays introduced to assist in assembling the sparse matrix could waste a lot of precious memory resources during the AMG setup stage. And, by using the improved bounds given in (11) and (13), we are able to use much shorter auxiliary arrays than the standard implementation in [1] and this can save a lot memory, especially when the bandwidth of the

Table 4 Number of iterations, wall-times (seconds), and OpenMP speedups of B_{TRIG}.

	1st			2nd			3nd			4nd		
N_T	#Iter	Time	Speedup	#Iter	Time	Speedup	#Iter	Time	Speedup	#Iter	Time	Speedup
1	49	41.69	—	49	41.48	—	48	40.96	—	44	37.75	—
2	49	23.42	1.78	48	22.93	1.81	48	22.87	1.79	44	21.25	1.78
4	49	16.67	2.50	49	16.62	2.50	48	16.30	2.51	44	15.37	2.46
8	49	14.30	2.91	48	13.94	2.98	48	13.91	2.95	44	12.92	2.92
12	48	14.00	2.98	48	13.99	2.97	47	13.58	3.02	44	12.99	2.91

sparse matrix A is small or the number of OpenMP threads is large. Let Length(M_P) be the total length of M_P, and let Length(M_A) be the total length of M_A. We compare these two auxiliary arrays (M_A and M_P) on the finest level as an example in Table 5. Numerical results show that this simple improvement can save about 87% storage when 12 threads are used on the finest level.

Table 5 Auxiliary memory storage on the finest level of the AMG setup for the pressure equation.

	Length(M_P)			Length(M_A)		
N_T	$N_T \times n$	L_P	Saving (%)	$N_T \times n$	L_A	Saving (%)
2	2,188,844	1,200,022	45.1	2,188,844	1,147,222	47.6
4	4,377,688	1,305,622	70.2	4,377,688	1,252,822	71.3
6	6,566,532	1,411,222	78.5	6,566,532	1,358,422	79.3
8	8,755,376	1,516,822	82.7	6,566,532	1,464,022	83.3
12	13,133,064	1,728,022	86.8	13,133,064	1,675,222	87.2

Acknowledgements The authors appreciate the anonymous referee for his or her suggestions which led to a better presentation of our method. The authors would like to thank RIPED, PetroChina, for providing the modified SPE10 test. Feng is partially supported by NSFC Grant 11201398. Shu is partially supported by NSFC Grants 91130002 and 11171281 and by the Scientific Research Fund of the Hunan Provincial Education Department of China #12A138. Xu is partially supported by NSFC Grant 91130011. Zhang is partially supported by the Dean's Startup Fund, Academy of Mathematics and System Sciences and by NSFC Grant 91130011.

References

1. hypre: A scalable linear solver library. URL `https://computation.llnl.gov/casc/linear_solvers/sls_hypre.html`
2. Al-Shaalan, T., Klie, H., Dogru, A., Wheeler, M.: Studies of Robust Two Stage Preconditioners for the Solution of Fully Implicit Multiphase Flow Problems. In: SPE Reservoir Simulation Symposium (2009)
3. Appleyard, J., Cheshire, I.: Nested factorization. In: SPE Reservoir Simulation Symposium (1983)
4. Appleyard, J., Cheshire, I., Pollard, R.: Special techniques for fully implicit simulators. In: Proceedings of the European Symposium on Enhanced Oil Recovery, Bournemouth, England, pp. 395–408 (1981)

5. Bank, R.E., Chan, T.F., Coughran Jr., W.M., Smith, R.K.: The alternate-block-factorization procedure for systems of partial differential equations. BIT **29**(4), 938–954 (1989)
6. Behie, A., Vinsome, P.: Block iterative methods for fully implicit reservoir simulation. Old SPE Journal **22**(5), 658–668 (1982)
7. Brandt, A., McCormick, S., Ruge, J.: Algebraic multigrid (AMG) for sparse matrix equations. In: Sparsity and its applications (Loughborough, 1983), pp. 257–284. Cambridge Univ. Press, Cambridge (1985)
8. Chen, Z., Huan, G., Ma, Y.: Computational methods for multiphase flows in porous media, vol. 2. Society for Industrial Mathematics (2006)
9. Christie, M., Blunt, M.: Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. SPE Reservoir Evaluation & Engineering **4**(4), 308–317 (2001)
10. Concus, P., Golub, G., Meurant, G.: Block preconditioning for the conjugate gradient method. SIAM J. Sci. Statist. Comput **6**(1) (1985)
11. Douglas, Jr., J., Peaceman, D.W., Rachford, D.: A method for calculating multi-dimensional displacement. Transaction of American Institute of Mining, Metallurgical, and Petroleum Engineers **216**, 297–306 (1959)
12. Falgout, R.: An introduction to algebraic multigrid. Computing in Science and Engineering **8**(6), 24 (2006)
13. Feng, C., Shu, S., Yue, X.: An Improvement for the OpenMP Version BoomerAMG. In: Proceedings of CCF HPC CHINA 2012, Zhangjiajie, China, pp. 321–328 (2012)
14. Hu, X., Liu, W., Qin, G., Xu, J., Yan, Y., Zhang, C.: Development of a fast auxiliary subspace pre-conditioner for numerical reservoir simulators. In: SPE Reservoir Characterization and Simulation Conference (2011)
15. Lacroix, S., Vassilevski, Y., Wheeler, J., Wheeler, M.: Iterative solution methods for modeling multiphase flow in porous media fully implicitly. SIAM J. Sci. Comput. **25**(3), 905–926 (electronic) (2003)
16. Lacroix, S., Vassilevski, Y.V., Wheeler, M.F.: Decoupling preconditioners in the implicit parallel accurate reservoir simulator (IPARS). Numer. Linear Algebra Appl. **8**(8), 537–549 (2001). Solution methods for large-scale non-linear problems (Pleasanton, CA, 2000)
17. Meyerink, J.: Iterative methods for the solution of linear equations based on incomplete block factorization of the matrix. In: SPE Reservoir Simulation Symposium (1983)
18. Oliker, L., Li, X., Husbands, P., Biswas, R.: Effects of Ordering Strategies and Programming Paradigms on Sparse Matrix Computations. SIAM Review **44**(3), 373–393 (2002)
19. Peaceman, D.W.: Presentation of a horizontal well in numerical reservoir simulation. In: The 11th SPE Symposium on Reservoir Simulation, SPE 21217 (1991)
20. Ruge, J.W., Stüben, K.: Algebraic multigrid. In: Multigrid methods, *Frontiers Appl. Math.*, vol. 3, pp. 73–130. SIAM, Philadelphia, PA (1987)
21. Saad, Y.: Iterative methods for sparse linear systems, second edn. Society for Industrial and Applied Mathematics, Philadelphia, PA (2003)
22. Stueben, K., Clees, T., Klie, H., Lu, B., Wheeler, M.: Algebraic multigrid methods (amg) for the efficient solution of fully implicit formulations in reservoir simulation. In: SPE Reservoir Simulation Symposium (2007)
23. Wallis, J.: Incomplete gaussian elimination as a preconditioning for generalized conjugate gradient acceleration. In: SPE Reservoir Simulation Symposium (1983)
24. Watts, J., Shaw, J.: A new method for solving the implicit reservoir simulation matrix equation. In: SPE Reservoir Simulation Symposium, 31 January-2 Feburary 2005, The Woodlands, Texas (2005)

Part II
Minisymposia

A FETI-DP algorithm for incompressible Stokes equations with continuous pressures

Xuemin Tu[1] and Jing Li[2]

1 Introduction

The FETI-DP algorithm was first extended to solving incompressible Stokes equations by Li [3], where a Dirichlet preconditioner was considered and the subdomain average pressure degrees of freedoms were selected as a primal constraint, in addition to the coarse level primal velocity constraints. The resulting coarse problem is a saddle point problem. The condition number bound is independent of the number of subdomains and grows only polylogarithmically with the size of the individual subdomain problems.

Recently, Kim, Lee, and Park [2] introduced a different FETI-DP formulation for this problem, where no pressure variables are selected as coarse level primal variables and the resulting coarse level problem is symmetric positive definite. Only the lumped preconditioner is considered in their paper.

Both approaches mentioned above are valid only for discretizations with a discontinuous pressure. Discontinuous pressures have also been used in domain decomposition algorithms for similar type saddle-point problems; see for example [1, 5, 7].

In this paper, we propose a FETI-DP algorithm for incompressible Stokes using either a lumped or a Dirichlet preconditioner with continuous pressure discretization; see also [4, 8] for more details. Our coarse problem includes no pressure variables and is symmetric positive definite.

2 Discretization and domain decomposition

The weak solution of the incompressible Stokes problem, on a bounded, two-dimensional polygonal domain Ω with a zero Dirichlet boundary condition, is given by: find $\mathbf{u}^* \in \left(H_0^1(\Omega)\right)^2 = \{\mathbf{v} \in (H^1(\Omega))^2 \mid \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega\}$ and $p^* \in L^2(\Omega)$, such that

$$\begin{cases} a(\mathbf{u}^*, \mathbf{v}) + b(\mathbf{v}, p^*) = (\mathbf{f}, \mathbf{v}), & \forall \mathbf{v} \in \left(H_0^1(\Omega)\right)^2, \\ b(\mathbf{u}^*, q) = 0, & \forall q \in L^2(\Omega), \end{cases} \tag{1}$$

where

[1]Department of Mathematics, University of Kansas, 1460 Jayhawk Blvd, Lawrence, KS 66045-7594, U.S.A. e-mail: xtu@math.ku.edu [2] Department of Mathematical Sciences, Kent State University, Kent, OH 44242, U.S.A. e-mail: li@emath.kent.edu

$$a(\mathbf{u}^*,\mathbf{v}) = \int_\Omega \nabla\mathbf{u}^* \cdot \nabla\mathbf{v}, \quad b(\mathbf{u}^*,q) = -\int_\Omega (\nabla\cdot\mathbf{u}^*)q, \quad (\mathbf{f},\mathbf{v}) = \int_\Omega \mathbf{f}\cdot\mathbf{v}.$$

We note that the solution of (1) is not unique, with the pressure p^* determined only up to an additive constant.

A Q_2-Q_1 Taylor-Hood mixed finite element is used in this paper to solve (1). The domain Ω is partitioned into shape-regular rectangular elements of characteristic size h. The pressure finite element space, $Q \subset L^2(\Omega)$, is taken as the space of continuous piecewise bilinear functions while the velocity finite element space, $\mathbf{W} \in \left(H_0^1(\Omega)\right)^2$, is formed by the continuous piecewise biquadratic functions.

The finite element solution $(\mathbf{u}, p) \in \mathbf{W} \bigoplus Q$ of (1) satisfies

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}, \tag{2}$$

where A, B, and $\mathbf{f}$ represent, respectively, the restrictions of $a(\cdot,\cdot)$, $b(\cdot,\cdot)$ and $(\mathbf{f},\cdot)$ to the finite-dimensional spaces $\mathbf{W}$ and Q. We use the same notation in this paper to represent both a finite element function and the vector of its nodal values. The solution of (2) always exists and is uniquely determined when the pressure is considered in the quotient space $Q/Ker(B^T)$, where $Ker(B^T)$ represents the kernel of B^T and is the space of constant pressures in Q. In this paper, when $q \in Q/Ker(B^T)$, q always has a zero average.

The Taylor-Hood mixed finite element space $\mathbf{W} \times Q$ is inf-sup stable in the sense that there exists a positive constant β, independent of h, such that, in matrix/vector form,

$$\sup_{\mathbf{w}\in\mathbf{W}} \frac{\langle q, B\mathbf{w}\rangle^2}{\langle \mathbf{w}, A\mathbf{w}\rangle} \geq \beta^2 \langle q, Zq\rangle, \qquad \forall q \in Q/Ker(B^T). \tag{3}$$

Here, as elsewhere in this paper, $\langle\cdot,\cdot\rangle$ represents the inner product of two vectors. The matrix Z represents the mass matrix defined on the pressure finite element space Q, i.e., for any $q \in Q$, $\|q\|_{L^2}^2 = \langle q, Zq\rangle$. It is easy to see, cf. [6, Lemma B.31], that Z is spectrally equivalent to $h^2 I$ for two-dimensional problems, where I represents the identity matrix of the same dimension, i.e., there exist positive constants c and C, such that

$$ch^2 I \leq Z \leq Ch^2 I. \tag{4}$$

Here, as in other places of this paper, c and C represent generic positive constants which are independent of the mesh size h and the subdomain diameter H (discussed below).

The domain Ω is decomposed into N non-overlapping polygonal subdomains Ω_i, $i = 1,2,\ldots,N$. Each subdomain is the union of a bounded number of elements, with the diameter of the subdomain on the order of H. The nodes on the interface of neighboring subdomains match across the subdomain boundaries $\Gamma = (\cup\partial\Omega_i)\backslash\partial\Omega$. Γ is composed of subdomain edges, which are regarded as open subsets of Γ, and of the subdomain vertices, which are end points of edges.

The velocity and pressure finite element spaces $\mathbf{W}$ and Q are decomposed into $\mathbf{W} = \mathbf{W}_I \bigoplus \mathbf{W}_\Gamma$, $Q = Q_I \bigoplus Q_\Gamma$, where $\mathbf{W}_I$ and Q_I are direct sums of independent

subdomain interior velocity spaces $\mathbf{W}_I^{(i)}$, and interior pressure spaces $Q_I^{(i)}$, respectively. $\mathbf{W}_\Gamma$ and Q_Γ are subdomain boundary velocity and pressure spaces, respectively. All functions in $\mathbf{W}_\Gamma$ and Q_Γ are continuous across the subdomain boundaries Γ; their degrees of freedom are shared by neighboring subdomains.

To formulate our algorithm, we introduce a partially sub-assembled subdomain boundary velocity space $\widetilde{\mathbf{W}}_\Gamma$,

$$\widetilde{\mathbf{W}}_\Gamma = \mathbf{W}_\Pi \bigoplus \mathbf{W}_\Delta = \mathbf{W}_\Pi \bigoplus \left(\bigoplus_{i=1}^{N} \mathbf{W}_\Delta^{(i)} \right).$$

Here $\mathbf{W}_\Pi$ is the continuous primal velocity space which forms the coarse level problem of the proposed algorithm. In this paper, we consider two choices of $\mathbf{W}_\Pi$. The first choice is with that $\mathbf{W}_\Pi$ is spanned by all the subdomain corner velocity nodal basis functions. In the second choice, $\mathbf{W}_\Pi$ is spanned by both subdomain corner velocity nodal basis functions and edge-average finite element basis functions. We note that the appropriate choice of $\mathbf{W}_\Pi$ depends on the preconditioner used in the algorithm. The first choice is sufficient for using the lumped preconditioner, but for the Dirichlet preconditioner the second one has to be used.

The space $\mathbf{W}_\Delta$ is the direct sum of subdomain dual interface velocity spaces $\mathbf{W}_\Delta^{(i)}$. The functions $\mathbf{w}_\Delta$ in $\mathbf{W}_\Delta$ are in general not continuous across Γ. In order to enforce their continuity, we construct a matrix B_Δ from $\{0, 1, -1\}$ such that for any $\mathbf{w}_\Delta$ in $\mathbf{W}_\Delta$, each row of $B_\Delta \mathbf{w}_\Delta = 0$ implies that the two independent degrees of freedom from the neighboring subdomains be the same. The range of B_Δ applied on $\mathbf{W}_\Delta$ is denoted by Λ, the vector space of the Lagrange multipliers. A positive scaling factor $\delta^\dagger(x)$ for each node x on the subdomain boundary Γ is defined as $\delta^\dagger(x) = 1/\mathcal{N}_x$, where $\mathcal{N}_x$ represents the number of subdomains sharing x. Multiplying the entries on each row of B_Δ by the corresponding scaling factor $\delta^\dagger(x)$ gives us $B_{\Delta,D}$.

The original linear system (2) is equivalent to: find $(\mathbf{u}_I,\ p_I,\ \mathbf{u}_\Delta,\ \mathbf{u}_\Pi,\ p_\Gamma,\ \lambda) \in \mathbf{W}_I \bigoplus Q_I \bigoplus \mathbf{W}_\Delta \bigoplus \mathbf{W}_\Pi \bigoplus Q_\Gamma \bigoplus \Lambda$, such that

$$\begin{bmatrix} A_{II} & B_{II}^T & A_{I\Delta} & A_{I\Pi} & B_{\Gamma I}^T & 0 \\ B_{II} & 0 & B_{I\Delta} & B_{I\Pi} & 0 & 0 \\ A_{\Delta I} & B_{I\Delta}^T & A_{\Delta\Delta} & A_{\Delta\Pi} & B_{\Gamma\Delta}^T & B_\Delta^T \\ A_{\Pi I} & B_{I\Pi}^T & A_{\Pi\Delta} & A_{\Pi\Pi} & B_{\Gamma\Pi}^T & 0 \\ B_{\Gamma I} & 0 & B_{\Gamma\Delta} & B_{\Gamma\Pi} & 0 & 0 \\ 0 & 0 & B_\Delta & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ p_I \\ \mathbf{u}_\Delta \\ \mathbf{u}_\Pi \\ p_\Gamma \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I \\ 0 \\ \mathbf{f}_\Delta \\ \mathbf{f}_\Pi \\ 0 \\ 0 \end{bmatrix}, \tag{5}$$

where the sub-blocks in the coefficient matrix represent the restrictions of A and B of (2) to appropriate subspaces. The leading three-by-three block can be ordered to become block diagonal with each diagonal block representing one independent subdomain problem.

Corresponding to the one-dimensional null space of (2), a basis of the one-dimensional null space of (5) has the form

$$\left(0,\, 1_{p_I},\, 0,\, 0,\, 1_{p_\Gamma},\, -B_{\Delta,D}[B_{I\Delta}^T \;\; B_{\Gamma\Delta}^T]\begin{bmatrix} 1_{p_I} \\ 1_{p_\Gamma}\end{bmatrix}\right), \tag{6}$$

where $1_{p_I} \in Q_I$ and $1_{p_\Gamma} \in Q_\Gamma$ represent vectors with each entry equal to 1.

3 A reduced symmetric positive semi-definite system

The system (5) can be reduced to a Schur complement problem for the variables $(p_\Gamma,\, \lambda)$. The leading four-by-four block of the coefficient matrix in (5) is invertible and the variables $(\mathbf{u}_I,\, p_I,\, \mathbf{u}_\Delta,\, \mathbf{u}_\Pi)$ can be eliminated and we obtain

$$G\begin{bmatrix} p_\Gamma \\ \lambda \end{bmatrix} = g, \tag{7}$$

where

$$G = \begin{bmatrix} B_{\Gamma I} & 0 & B_{\Gamma\Delta} & B_{\Gamma\Pi} \\ 0 & 0 & B_\Delta & 0 \end{bmatrix} \begin{bmatrix} A_{II} & B_{II}^T & A_{I\Delta} & A_{I\Pi} \\ B_{II} & 0 & B_{I\Delta} & B_{I\Pi} \\ A_{\Delta I} & B_{I\Delta}^T & A_{\Delta\Delta} & A_{\Delta\Pi} \\ A_{\Pi I} & B_{I\Pi}^T & A_{\Pi\Delta} & A_{\Pi\Pi} \end{bmatrix}^{-1} \begin{bmatrix} B_{\Gamma I}^T & 0 \\ 0 & 0 \\ B_{\Gamma\Delta}^T & B_\Delta^T \\ B_{\Gamma\Pi}^T & 0 \end{bmatrix}, \tag{8}$$

and

$$g = \begin{bmatrix} B_{\Gamma I} & 0 & B_{\Gamma\Delta} & B_{\Gamma\Pi} \\ 0 & 0 & B_\Delta & 0 \end{bmatrix} \begin{bmatrix} A_{II} & B_{II}^T & A_{I\Delta} & A_{I\Pi} \\ B_{II} & 0 & B_{I\Delta} & B_{I\Pi} \\ A_{\Delta I} & B_{I\Delta}^T & A_{\Delta\Delta} & A_{\Delta\Pi} \\ A_{\Pi I} & B_{I\Pi}^T & A_{\Pi\Delta} & A_{\Pi\Pi} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}_I \\ 0 \\ \mathbf{f}_\Delta \\ \mathbf{f}_\Pi \end{bmatrix}. \tag{9}$$

We denote

$$\widetilde{A} = \begin{bmatrix} A_{II} & B_{II}^T & A_{I\Delta} & A_{I\Pi} \\ B_{II} & 0 & B_{I\Delta} & B_{I\Pi} \\ A_{\Delta I} & B_{I\Delta}^T & A_{\Delta\Delta} & A_{\Delta\Pi} \\ A_{\Pi I} & B_{I\Pi}^T & A_{\Pi\Delta} & A_{\Pi\Pi} \end{bmatrix} \quad \text{and} \quad B_C = \begin{bmatrix} B_{\Gamma I} & 0 & B_{\Gamma\Delta} & B_{\Gamma\Pi} \\ 0 & 0 & B_\Delta & 0 \end{bmatrix}. \tag{10}$$

It is easy to see that $-G$ is the Schur complement of the coefficient matrix of (5) with respect to the last two row blocks. By the Sylvester law of inertia, G is symmetric positive semi-definite. The null space of G can be derived from the null space of the original coefficient matrix of (5), and its basis has the form, cf. (6),

$$\left(1_{p_\Gamma},\, -B_{\Delta,D}[B_{I\Delta}^T \;\; B_{\Gamma\Delta}^T]\begin{bmatrix} 1_{p_I} \\ 1_{p_\Gamma}\end{bmatrix}\right).$$

Let $X = Q_\Gamma \bigoplus \Lambda$. The range of G, denoted by R_G, is the subspace of X, which is orthogonal to the null space of G and has the form

$$R_G = \left\{ \begin{bmatrix} g_{p_\Gamma} \\ g_\lambda \end{bmatrix} \in X \;\Big|\; g_{p_\Gamma}^T 1_{p_\Gamma} - g_\lambda^T \left(B_{\Delta,D} [B_{I\Delta}^T \;\; B_{\Gamma\Delta}^T] \begin{bmatrix} 1_{p_I} \\ 1_{p_\Gamma} \end{bmatrix} \right) = 0 \right\}. \qquad (11)$$

The restriction of G to its range R_G is positive definite.

The main operation in the implementation of multiplying G by a vector is the product of $\widetilde{A}^{-1}$ with a vector, cf. (8) and (9). We denote

$$A_{rr} = \begin{bmatrix} A_{II} & B_{II}^T & A_{I\Delta} \\ B_{II} & 0 & B_{I\Delta} \\ A_{\Delta I} & B_{I\Delta}^T & A_{\Delta\Delta} \end{bmatrix}, \;\; A_{\Pi r} = A_{r\Pi}^T = \begin{bmatrix} A_{\Pi I} & B_{I\Pi}^T & A_{\Pi\Delta} \end{bmatrix}, \;\; f_r = \begin{bmatrix} \mathbf{f}_I \\ 0 \\ \mathbf{f}_\Delta \end{bmatrix},$$

and define the Schur complement

$$S_\Pi = A_{\Pi\Pi} - A_{\Pi r} A_{rr}^{-1} A_{r\Pi}.$$

By the Sylvester law of inertia, S_Π is symmetric positive definite and defines the coarse level problem in the algorithm. The product

$$\begin{bmatrix} A_{II} & B_{II}^T & A_{I\Delta} & A_{I\Pi} \\ B_{II} & 0 & B_{I\Delta} & B_{I\Pi} \\ A_{\Delta I} & B_{I\Delta}^T & A_{\Delta\Delta} & A_{\Delta\Pi} \\ A_{\Pi I} & B_{I\Pi}^T & A_{\Pi\Delta} & A_{\Pi\Pi} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}_I \\ 0 \\ \mathbf{f}_\Delta \\ \mathbf{f}_\Pi \end{bmatrix}$$

can then be represented by

$$\begin{bmatrix} A_{rr}^{-1} f_r \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -A_{rr}^{-1} A_{r\Pi} \\ I_\Pi \end{bmatrix} S_\Pi^{-1} \left(\mathbf{f}_\Pi - A_{\Pi r} A_{rr}^{-1} f_r \right),$$

which requires solving the coarse level problem once and independent subdomain Stokes problems with Neumann type boundary conditions twice.

4 Preconditioners and condition number bounds

Both the lumped and the Dirichlet preconditioners are proposed here for solving (7). We define

$$\widetilde{V} = \mathbf{W}_I \bigoplus Q_I \bigoplus \mathbf{W}_\Delta \bigoplus \mathbf{W}_\Pi,$$

and its subspace

$$\widetilde{V}_0 = \left\{ w = (\mathbf{w}_I,\; p_I,\; \mathbf{w}_\Delta,\; \mathbf{w}_\Pi) \in \widetilde{V} : B_{II}\mathbf{w}_I + B_{I\Delta}\mathbf{w}_\Delta + B_{I\Pi}\mathbf{w}_\Pi = 0 \right\}.$$

We note that $\langle \cdot,\cdot \rangle_{\widetilde{A}}$ defines an inner product on $\widetilde{V}_0$. We denote the restriction operator from $\widetilde{V}$ onto $\mathbf{W}_\Delta$ by $\widetilde{R}_\Delta$ such that for any $v = (\mathbf{w}_I,\ p_I,\ \mathbf{w}_\Delta,\ \mathbf{w}_\Pi) \in \widetilde{V}$, $\widetilde{R}_\Delta v = \mathbf{w}_\Delta$.

The lumped preconditioner is given by

$$M_L^{-1} = \begin{bmatrix} \frac{1}{h^2} I_{p_\Gamma} & \\ & M_{L,\lambda}^{-1} \end{bmatrix},$$

where I_{p_Γ} is the identity matrix of the same length as p_Γ and $M_{L,\lambda}^{-1} = B_{\Delta,D}\widetilde{R}_\Delta \widetilde{A} \widetilde{R}_\Delta^T B_{\Delta,D}^T$.

Let $M_{D,\lambda}^{-1} = B_{\Delta,D} H_\Delta B_{\Delta,D}^T$. Then the Dirichlet preconditioner is defined as

$$M_D^{-1} = \begin{bmatrix} \frac{1}{h^2} I_{p_\Gamma} & \\ & M_{D,\lambda}^{-1} \end{bmatrix},$$

where H_Δ is the direct sum of the discrete subdomain harmonic extension operators.

The following lemma is used for obtaining the upper bound estimate in Theorem 1, and it is valid for both preconditioners, denoted here by M^{-1}.

Lemma 1. *For any $v \in \widetilde{V}_0$, $\left\langle M^{-1} B_C v, B_C v \right\rangle \le \Phi(H,h) \left\langle \widetilde{A} v, v \right\rangle$. Here, for the lumped preconditioner, $\Phi(H,h) = C(H/h)(1+\log(H/h))$ with only corner variables in the coarse space; $\Phi(H/h) = C(H/h)$ with both corner and edge-average variables. For the Dirichlet preconditioner, $\Phi(H,h) = C(1+\log(H/h))^2$ with both corner and edge-average coarse variables.*

The second lemma is used for the lower bound estimate. For the lumped preconditioner, the corner primal constraints are sufficient for the coarse space to prove this lemma. However, for the Dirichlet preconditioner, both corner and edge-average constraints have to be included in the coarse space.

Lemma 2. *For any given $y = (g_{p_\Gamma}, g_\lambda) \in R_G$, there exits $v \in \widetilde{V}_0$, such that $B_C v = y$, and $\left\langle \widetilde{A} v, v \right\rangle \le \frac{C}{\beta^2} \left\langle M^{-1} y, y \right\rangle$.*

Theorem 1. *For all $x = (p_\Gamma, \lambda) \in R_{M^{-1}G}$,*

$$c\beta^2 \langle Mx, x \rangle \le \langle Gx, x \rangle \le \Phi(H,h) \langle Mx, x \rangle,$$

where $\Phi(H,h)$ is as defined in Lemma 1 and β is the inf-sup constant of (3).

5 Numerical experiments

We solve the incompressible Stokes problem in the square domain $\Omega = [0,1] \times [0,1]$. Zero Dirichlet boundary conditions are used. The right-hand side function $\mathbf{f}$ is chosen such that the exact solution is

Table 1 Performance with the lumped preconditioner M_L^{-1}.

		Vertex			Vertex and edge		
H/h	#sub	λ_{min}	λ_{max}	iter	λ_{min}	λ_{max}	iter
8	4×4	0.31	32.28	31	0.31	4.30	19
	8×8	0.31	37.25	46	0.31	4.50	20
	16×16	0.31	38.40	51	0.31	4.53	21
	24×24	0.31	38.62	51	0.31	4.55	21
	32×32	0.31	38.68	51	0.31	4.55	21
#sub	H/h	λ_{min}	λ_{max}	iter	λ_{min}	λ_{max}	iter
8×8	4	0.30	15.92	34	0.30	3.21	18
	8	0.31	37.25	46	0.30	4.50	20
	12	0.31	60.62	56	0.31	6.65	24
	16	0.31	85.32	62	0.31	8.87	27
	24	0.31	137.49	73	0.31	13.40	32

$$\mathbf{u} = \begin{bmatrix} \sin^3(\pi x)\sin^2(\pi y)\cos(\pi y) \\ -\sin^2(\pi x)\sin^3(\pi y)\cos(\pi x) \end{bmatrix} \quad \text{and} \quad p = x^2 - y^2.$$

The Q_2-Q_1 Taylor-Hood mixed finite element is used for the finite element solution. The preconditioned system is solved by a CG iteration; the iteration is stopped when the L^2-norm of the residual is reduced by a factor of 10^{-6}.

Table 1 shows the minimum and maximum eigenvalues of the iteration matrix $M_L^{-1}G$, and the iteration counts. Two different coarse level spaces are tested in the experiments: the coarse space spanned by only the subdomain corner velocities, and the coarse space spanned by both the subdomain corner and the subdomain edge-average velocities. The additional edge-average velocity components in the coarse level problem improve the convergence rate even though they are not necessary for the analysis.

Table 2 shows the performance of our algorithm for solving the same problem with the Dirichlet preconditioner. For this case, the additional edge-average velocity components included in the coarse level space are necessary, which is consistent with our theory.

Acknowledgements This work was supported in part by National Science Foundation Contract No. DMS-1115759.

Table 2 Performance with the Dirichlet preconditioner M_D^{-1}.

		Vertex			Vertex and edge		
H/h	#sub	λ_{min}	λ_{max}	iter	λ_{min}	λ_{max}	iter
8	4×4	0.30	4.40	18	0.30	3.04	17
	8×8	0.29	5.03	24	0.30	3.50	18
	16×16	0.26	5.28	25	0.30	3.92	19
	24×24	0.24	5.33	25	0.30	4.10	19
	32×32	0.23	5.36	25	0.30	4.18	19
#sub	H/h	λ_{min}	λ_{max}	iter	λ_{min}	λ_{max}	iter
8×8	4	0.27	4.15	21	0.30	3.15	17
	8	0.29	5.03	24	0.30	3.50	18
	12	0.29	5.60	25	0.30	3.92	18
	16	0.30	6.04	25	0.30	4.24	18
	24	0.30	6.70	26	0.30	4.71	19

References

1. Paulo Goldfeld, Luca F. Pavarino, and Olof B. Widlund. Balancing Neumann-Neumann preconditioners for mixed approximations of heterogeneous problems in linear elasticity. *Numer. Math.*, 95(2):283–324, 2003.
2. Hyea Hyun Kim, Chang-Ock Lee, and Eun-Hee Park. A FETI-DP formulation for the Stokes problem without primal pressure components. *SIAM J. Numer. Anal.*, 47(6):4142–4162, 2010.
3. Jing Li. A Dual-Primal FETI method for incompressible Stokes equations. *Numer. Math.*, 102:257–275, 2005.
4. Jing Li and Xuemin Tu. A non-overlapping domain decomposition method for incompressible Stokes equation with continuous pressure. *SIAM J. Numer. Anal.*, in press. Available at http://arxiv.org/abs/1204.1899.
5. Jing Li and Olof B. Widlund. BDDC algorithms for incompressible Stokes equations. *SIAM J. Numer. Anal.*, 44(6):2432–2455, 2006.
6. Andrea Toselli and Olof B. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer Verlag, Berlin-Heidelberg-New York, 2005.
7. Xuemin Tu. A BDDC algorithm for a mixed formulation of flows in porous media. *Electron. Trans. Numer. Anal.*, 20:164–179, 2005.
8. Xuemin Tu and Jing Li. A unified FETI-DP approach for incompressible Stokes equations. *Internat. J. Numer. Methods Engrg.*, 94(2):111–220, 2013.

Generating Equidistributed Meshes in 2D via Domain Decomposition

Ronald D. Haynes[1] and Alexander J. M. Howse[1]

1 Introduction

There are many occasions when the use of a uniform spatial grid would be prohibitively expensive for the numerical solution of partial differential equations (PDEs). In such situations, a popular strategy is to generate an adaptive mesh by either varying the number of mesh points, the order of the numerical method, or the location of mesh points throughout the domain, in order to best resolve the solution. It is the latter of these options, known as *moving mesh methods*, which is our focus. In this case the physical PDE of interest is coupled with equations which adjust the position of mesh points to best "equidistribute" a particular measure of numerical error. This coupled system of equations is solved to generate the solution and the corresponding mesh simultaneously, see [7] for a recent overview.

A simple method for adaptive grid generation in two spatial dimensions is outlined in [8] by Huang and Sloan, in which a finite difference two dimensional adaptive mesh method is developed by applying a variation of de Boor's equidistribution principle (EP) [1, 2]. The equidistribution principle states that an appropriately chosen mesh should equally distribute some measure of the solution variation or computational error over the entire domain. Mackenzie [9] extends upon the work of [8] by presenting a finite volume discretization of the mesh equations, as well as an efficient iterative approach for solving these equations, referred to as "an alternating line Gauss-Seidel relaxation approach".

In this paper, we propose a parallel domain decomposition (DD) solution of the 2D adaptive method of [8]. In Section 2 we review the derivation of the mesh PDEs of [8] and discuss possible boundary conditions. In Sections 3 and 4 we present classical and optimized Schwarz methods for the generation of 2D equidistributed meshes, and in Section 5 we describe the numerical implementation of this approach and provide numerical results.

2 2D Mesh Generation

To begin, we review the derivation of the equations which govern mesh equidistribution in two spatial dimensions from [8], defining a mesh in the physical variables (x,y) which *best* resolves a given function $u(x,y)$. Let $\mathbf{x} = [x,y]^T$ be the spatial coor-

[1]Memorial University of Newfoundland, Department of Mathematics & Statistics, St. John's, NL, Canada A1C 5S7 e-mail: {rhaynes}{z37ajmh}@mun.ca

dinates of a mesh in a 2D physical domain, Ω_p. We introduce the coordinate transformation $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$, where $\boldsymbol{\xi} = [\xi, \eta]^T$ denotes the spatial coordinates on the computational domain, $\Omega_c = [0,1] \times [0,1]$. Here we determine a mesh which equidistributes the arc-length of $u(x,y)$ over Ω_p. The scaled arc-length variation of u along the arc element from $\mathbf{x}$ to $\mathbf{x} + d\mathbf{x}$ can be expressed as

$$ds = [a^2(du)^2 + d\mathbf{x}^T d\mathbf{x}]^{1/2} = [d\mathbf{x}^T \boldsymbol{M} d\mathbf{x}]^{1/2}, \tag{1}$$

where $\boldsymbol{M} = a^2 \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} u^T + \boldsymbol{I}$ and $a \in [0,1]$ is a user specified relaxation parameter. The extreme cases are $a = 0$, which produces a uniform mesh, and $a = 1$, which produces a mesh equidistributing the arc-length monitor function. Making use of the mesh transformation $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$, (1) can be expressed as

$$ds = [d\boldsymbol{\xi}^T \boldsymbol{J}^T \boldsymbol{M} \boldsymbol{J} d\boldsymbol{\xi}]^{1/2}, \tag{2}$$

where $\boldsymbol{J}$ is the Jacobian of the transformation.

The equidistribution principle follows from (2): if $u(\mathbf{x}(\boldsymbol{\xi}))$ is to have the same value ds along any arc element in the computational domain with fixed length $[d\boldsymbol{\xi}^T d\boldsymbol{\xi}]^{1/2}$, then (2) must be independent of the coordinate $\boldsymbol{\xi}$. This implies that $\boldsymbol{J}^T \boldsymbol{M} \boldsymbol{J}$ should be independent of $\boldsymbol{\xi}$, or

$$[d\boldsymbol{\xi}^T \boldsymbol{J}^T \boldsymbol{M} \boldsymbol{J} d\boldsymbol{\xi}]^{1/2} = [d\boldsymbol{\xi}^T \widetilde{\boldsymbol{M}} d\boldsymbol{\xi}]^{1/2}, \tag{3}$$

where $\widetilde{\boldsymbol{M}}$ is a constant and hence $\boldsymbol{\xi}$-independent matrix. If a coordinate transformation can be found which satisfies (3), u will have the same variation at any point in Ω_p along any arc of length

$$\left[\left(\frac{\partial \mathbf{x}}{\partial \xi} d\xi + \frac{\partial \mathbf{x}}{\partial \eta} d\eta \right)^T \left(\frac{\partial \mathbf{x}}{\partial \xi} d\xi + \frac{\partial \mathbf{x}}{\partial \eta} d\eta \right) \right]^{1/2}.$$

A transformation satisfying (3) for some matrix $\widetilde{\boldsymbol{M}}$ will be called an *equidistribution*, and (3) an *equidistribution principle*.

Usually (3) cannot be satisfied by the coordinate transformation on the whole computational domain. However, if (3) is weakened so that the transformation is only required to satisfy (3) locally; that is, we only require $\widetilde{\boldsymbol{M}}$ to be constant along a given coordinate line, it is possible to find a local equidistribution on Ω_p. In 2D this leads to the system:

$$\left[\begin{pmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \end{pmatrix}^T \boldsymbol{M} \begin{pmatrix} \frac{\partial x}{\partial \xi} \\ \frac{\partial y}{\partial \xi} \end{pmatrix} \right]^{1/2} = c_1(\eta), \qquad \left[\begin{pmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \end{pmatrix}^T \boldsymbol{M} \begin{pmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \end{pmatrix} \right]^{1/2} = c_2(\xi), \tag{4}$$

where $c_1(\eta)$ and $c_2(\xi)$ are constant in the ξ and η directions respectively. These constants are eliminated by numerical differencing.

Instead of using the scaled arc-length matrix $\boldsymbol{M}$, in practice we modify $\boldsymbol{M}$ as $\boldsymbol{M} = k \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} u^T + \boldsymbol{I}$, where $k = a^2/(1 + b \boldsymbol{\nabla} u^T \boldsymbol{\nabla} u)$. The parameter $b \geq 0$ is used

to prevent problems where extremely small mesh spacing or mesh tangling could occur, that is when $|\nabla u|$ is very large.

System (4) will determine the internal mesh points. In [8] a combination of Dirichlet and Neumann conditions are used along $\partial\Omega_c$:

$$x(0,\eta) = y(\xi,0) = 0, \quad x(1,\eta) = y(\xi,1) = 1, \tag{5}$$

$$\frac{\partial x}{\partial \eta}(\xi,0) = \frac{\partial x}{\partial \eta}(\xi,1) = \frac{\partial y}{\partial \xi}(0,\eta) = \frac{\partial y}{\partial \xi}(1,\eta) = 0, \tag{6}$$

where $\xi,\eta \in [0,1]$. The Dirichlet conditions are consistent with the requirement that there are mesh points on the boundary of the domain. The Neumann orthogonality conditions are arbitrary, and in fact can cause smoothness issues near the domain boundaries. As an alternative, we follow [9] and apply the 1D EP,

$$(M(x)x_\xi)_\xi = 0, \quad x(0) = 0, \quad x(1) = 1, \tag{7}$$

to determine $x(\xi,0)$, $x(\xi,1)$, $y(0,\eta)$ and $y(1,\eta)$. The 1D analog of the system (4), given in (7), has previously been solved by DD methods in [3, 5, 6].

3 Classical Schwarz Domain Decomposition

For the two dimensional mesh adaptation problem, the computational domain $\Omega_c = [0,1]\times[0,1]$, can either be decomposed in just the ξ or just the η directions, or in both directions. This results in "strip" or "block" configurations of subdomains respectively. Here we discuss DD applied in the ξ direction only. That is, we decompose the ξ interval $[0,1]$ into subintervals $[\alpha_\xi^i, \beta_\xi^i]$, $i = 1,\ldots,S$, where $\alpha_\xi^1 = 0$, $\beta_\xi^S = 1$, and we assume the subintervals satisfy the overlap conditions:

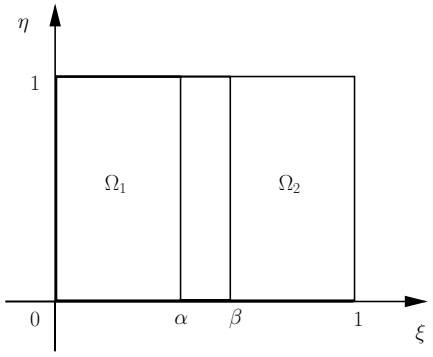


Fig. 1 DD in ξ using in 2 subdomains.

$$\alpha_\xi^i < \alpha_\xi^{i+1} < \beta_\xi^i < \beta_\xi^{i+1}.$$

The resulting decomposition has S subdomains, denoted by $\Omega_i = [\alpha_\xi^i, \beta_\xi^i]\times[0,1]$ for $i = 1,\ldots,S$. The boundary conditions (5–6) or (7) are used along the ends of each strip and transmission conditions are specified along the newly created interfaces.

Consider the 2D adaptive mesh system, (4), for the $S = 2$ case. We split Ω_c into subdomains Ω_1 and Ω_2 as in Figure 1. Let x_i^n denote the subdomain solution on Ω_i, for $i = 1,2$. We consider the following DD iteration: for $n = 1,2,\ldots$, solve

$$\left[\begin{pmatrix} \frac{\partial x_i^n}{\partial \xi} \\ \frac{\partial y_i^n}{\partial \xi} \end{pmatrix}^T \boldsymbol{M}(x_i^n, y_i^n) \begin{pmatrix} \frac{\partial x_i^n}{\partial \xi} \\ \frac{\partial y_i^n}{\partial \xi} \end{pmatrix}\right]^{1/2} = c_1(\eta), \tag{8}$$

$$\left[\begin{pmatrix} \frac{\partial x_i^n}{\partial \eta} \\ \frac{\partial y_i^n}{\partial \eta} \end{pmatrix}^T \boldsymbol{M}(x_i^n, y_i^n) \begin{pmatrix} \frac{\partial x_i^n}{\partial \eta} \\ \frac{\partial y_i^n}{\partial \eta} \end{pmatrix}\right]^{1/2} = c_2(\xi), \tag{9}$$

for $i = 1,2$ and $\boldsymbol{\xi} \in \Omega_i$. The classical Schwarz method uses the transmission conditions

$$x_1^n(\beta,\eta) = x_2^{n-1}(\beta,\eta), \quad y_1^n(\beta,\eta) = y_2^{n-1}(\beta,\eta), \tag{10}$$

$$x_2^n(\alpha,\eta) = x_1^{n-1}(\alpha,\eta), \quad y_2^n(\alpha,\eta) = y_1^{n-1}(\alpha,\eta). \tag{11}$$

On $\partial(\Omega_c \cap \Omega_i)$ the boundary conditions (5) are used, along with the 1D EP to determine $x(\xi,0)$, $x(\xi,1)$, $y(0,\eta)$ and $y(1,\eta)$.

Each DD iteration requires a pair of PDEs to be solved, each a "smaller" version of the local EP (4). These problems are solved in an iterative manner: given initial approximations to be used along interfaces, the PDEs (8–9) are solved, and then solution information along the interfaces is exchanged between subdomains. The PDEs are then solved again, now with updated boundary data, and the process repeats. By iterating, the subdomain solutions converge to the desired solution $\mathbf{x}$ on their respective subdomains. As is well known, classical Schwarz requires the subdomains to overlap [4].

4 Optimized Boundary Conditions

Classical Schwarz is known to converge slowly. As a way to remedy this, we propose the use of higher order, Robin type, transmission conditions along the artificial interfaces. As before, we decompose $\Omega_c = [0,1] \times [0,1]$ into subdomains $\Omega_1 = [0,\beta] \times [0,1]$ and $\Omega_2 = [\alpha,1] \times [0,1]$, where $\alpha \leq \beta$.

We define, for any differentiable functions $x(\xi,\eta)$ and $y(\xi,\eta)$, the operators

$$B_1(x) = x_\xi + px, \qquad B_2(x) = x_\xi - px,$$
$$B_3(x,y) = S_1(x,y) + px, \quad B_4(x,y) = S_1(x,y) - px,$$

where

$$S_1(x,y) = \sqrt{\begin{pmatrix} x_\xi \\ y_\xi \end{pmatrix}^T M \begin{pmatrix} x_\xi \\ y_\xi \end{pmatrix}}, \quad M = \frac{a^2 w \cdot w^T}{1 + b w^T \cdot w} + I$$

and

$$w = \frac{1}{x_\xi y_\eta - x_\eta y_\xi} \left[u_\xi y_\eta - u_\eta y_\xi, -u_\xi x_\eta + u_\eta x_\xi\right]^T.$$

We propose two possible sets of transmission conditions. The first are simple linear Robin conditions using the derivative normal to the artificial boundaries:

$$\begin{aligned} B_1(x_1^n(\beta,\eta)) &= B_1(x_2^{n-1}(\beta,\eta)), \quad B_1(y_1^n(\beta,\eta)) = B_1(y_2^{n-1}(\beta,\eta)) \\ B_2(x_2^n(\alpha,\eta)) &= B_2(x_1^{n-1}(\alpha,\eta)), \quad B_2(y_2^n(\alpha,\eta)) = B_2(y_1^{n-1}(\alpha,\eta)). \end{aligned} \tag{12}$$

The second set are of nonlinear Robin type, similar to those used in an optimized Schwarz algorithm for 1D mesh generation in [3]. We replace the x equations of (12) by

$$\begin{aligned} B_3(x_1^n(\beta,\eta),y_1^n(\beta,\eta)) &= B_3(x_2^{n-1}(\beta,\eta)y_2^{n-1}(\beta,\eta)) \\ B_4(x_2^n(\alpha,\eta),y_2^n(\alpha,\eta)) &= B_4(x_1^{n-1}(\alpha,\eta),y_1^{n-1}(\alpha,\eta)). \end{aligned} \tag{13}$$

Note, the mesh PDE (8) indicates that the nonlinear term S_1 in the operator B_3 is constant across the $\xi = \alpha$ and $\xi = \beta$ interfaces. Furthermore, as the system of equations resulting from (8-9) are already nonlinear, the nonlinear transmission conditions will not have a large impact on the cost of solving the system.

5 Numerical Implementation and Results

The local EP (4), the physical boundary conditions on Ω_c, and the transmission conditions (10, 11), (12) or (13), are discretized using standard finite differences on a uniform grid in the computational (ξ,η) variables. Second order centered differences are used, using the ghost value technique as needed at the boundaries to ensure the scheme is second order. The nonlinear transmission conditions require nonlinear, rather than linear, equations to be solved at the interface. This is not onerous as the whole system is solved with a Newton iteration.

In the examples we use the test function $u(x,y) = \left[1-e^{15(x-1)}\right]\sin(\pi y)$. The function is shown, along with its locally equidistributed mesh, in Figures 2 and 3. The physical mesh (x,y) is generated by solving (4) using a grid of 18×18 uniformly spaced mesh points in Ω_c. For this example, we use an optimized Schwarz iteration, with transmission conditions (12), on 2 subdomains with 4 points of overlap in the ξ direction. Here the number of points of overlap refers to the number of shared grid points, the overlap width is approximately half of this number times $\Delta\xi$. We choose the parameters $a = 0.7$, $b = 0.05$ and $p = 2.3$. The mesh on subdomain 1 is shown in red, on subdomain 2 in blue, and the overlap region in purple. In general, the meshes obtained by the different methods will be visually indistinguishable from one another at convergence. To compare the DD methods we will plot their convergence histories.

In Figure 4 we plot the maximum error between the subdomain and single domain solutions for each of x_1^n, x_2^n, y_1^n, and y_2^n obtained using classical Schwarz. These are obtained over a 12 by 12 grid with 4 points of overlap in ξ and parameters

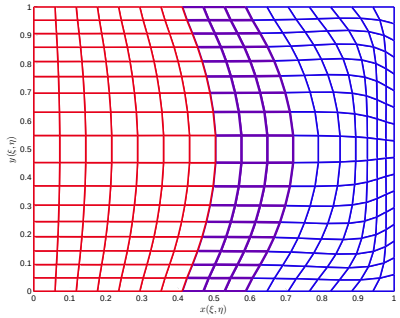


Fig. 2 Adaptive mesh generated for the test function.

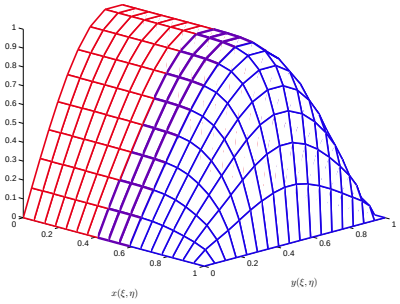


Fig. 3 The test function plotted using an adaptive mesh.

$a = 0.7$ and $b = 0.05$. As can be seen, each component of the solution converges at approximately the same rate, so we simplify our discussion by comparing the convergence of only x_1^n in the remaining figures.

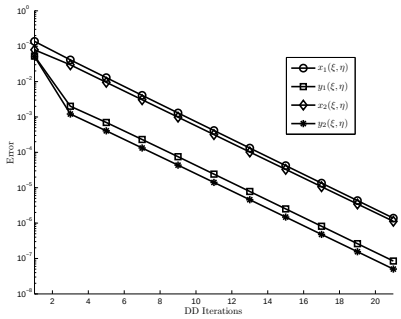


Fig. 4 Classical Schwarz convergence histories for each part of the solution, $x_{1,2}^n$ and $y_{1,2}^n$.

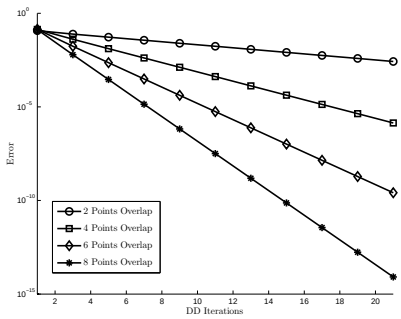


Fig. 5 Classical Schwarz convergence histories for varying amounts of overlap.

In Figure 5 we compare the classical Schwarz algorithm for varying amounts of overlap, using 2, 4, 6 and 8 points of overlap in the ξ direction. As expected, the rate of convergence improves as the overlap increases.

For the two possible optimized Schwarz iterations, we examine the effect of varying the parameter p in Figures 6 and 7. To generate these results we use a 12 by 12 mesh with two points of overlap in the ξ direction and parameters $a = 0.7$ and $b = 0.05$. For both types of transmission conditions, the best performance observed occurs for $p = 2$. Comparing the linear Robin condition (Figure 6) and nonlinear Robin condition (Figure 7), we see that the convergence histories for a general p are very similar. To examine these similarities, we plot the convergence histories for both optimized iterations for $p = 1, 2, 3$ on the same set of axes in Figure 8. We see that while the variations in this particular case are small, the nonlinear trans-

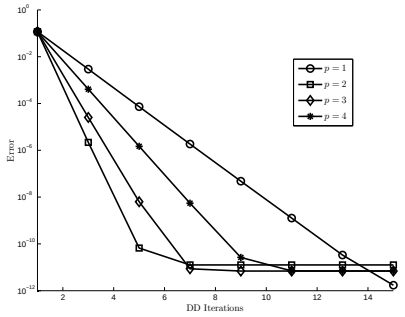


Fig. 6 Convergence histories for the Schwarz iteration using linear Robin conditions for varying p.

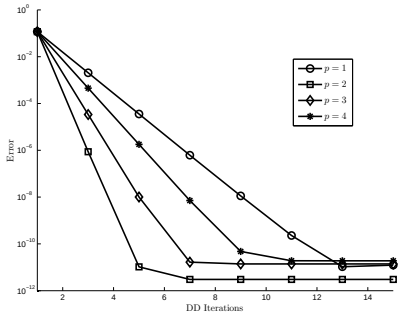


Fig. 7 Convergence histories for the Schwarz iteration using nonlinear Robin conditions for varying p.

mission conditions consistently outperform the linear Robin conditions. This is also observed in the results of Figure 9, in which we plot convergence histories for all three proposed DD algorithms. For this example we use a 12 by 12 mesh decomposed into two subdomains, with two points of overlap in ξ and parameters $a = 0.7$ and $b = 0.05$. In this example we see that both optimized Schwarz methods vastly outperform classical Schwarz, with the nonlinear transmission conditions slightly outperforming the linear Robin conditions.

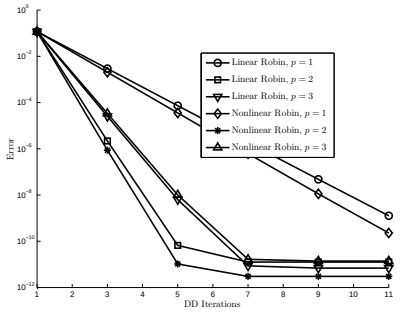


Fig. 8 Convergence histories for linear and nonlinear Robin transmission conditions with varying p.

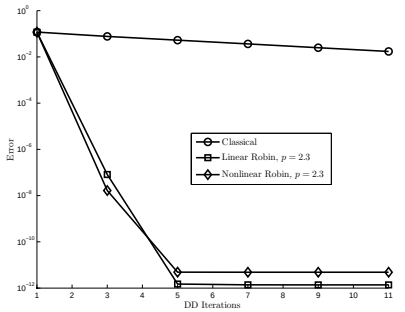


Fig. 9 Convergence histories for all three iterations considered.

Another way to assess the meshes obtained from a DD iteration is to compute a mesh quality measure. An equidistribution quality measure for each element K of the grid, $Q_{eq}(K)$, is presented in [7]. The maximum of Q_{eq} over all elements is 1 if and only if the equidistribution condition is satisfied exactly. The larger the value of $max_K Q_{eq}(K)$ the farther the mesh is from equidistributing $\boldsymbol{M}$. In Table 1 we compute the $max_K Q_{eq}(K)$ for the first five iterations of each proposed Schwarz algorithm. The zero column gives the mesh quality measure for the initial uniform 12×12

mesh and the ∞ column gives the mesh quality measure for the mesh obtained by solving system (4) over a single domain. Note, local equidistribution will not give a value of 1 for the mesh quality measure. We see that the meshes obtained by the optimized Schwarz algorithms rapidly give good meshes.

Table 1 Mesh quality measures for the grids obtained by the proposed Schwarz iterations.

Iterations	0	1	2	3	4	5	∞
Classical	1.6375	1.3630	1.3629	1.3178	1.3136	1.2795	1.1979
Linear Robin	1.6375	2.0076	1.1979	1.1979	1.1979	1.1979	1.1979
Nonlinear Robin	1.6375	2.0114	1.1979	1.1979	1.1979	1.1979	1.1979

6 Conclusion

In summary, we have proposed three different Schwarz DD iterations for obtaining 2D adaptive meshes defined by a local equidistribution principle. The numerical results show that the optimized methods provide a significant improvement over the slow convergence of classical Schwarz, with the nonlinear transmission conditions inspired by the work of [3] exhibiting the best results. Ongoing work includes the theoretical analysis of these DD approaches for 2D mesh generation and coupling the DD mesh generation with a DD solver for the physical PDE of interest.

References

1. de Boor, C.: Good approximation by splines with variable knots. In: Spline functions and approximation theory (Proc. Sympos., Univ. Alberta, Edmonton, Alta., 1972), pp. 57–72. Internat. Ser. Numer. Math., Vol. 21. Birkhäuser, Basel (1973)
2. de Boor, C.: Good approximation by splines with variable knots. II. In: Conference on the Numerical Solution of Differential Equations (Univ. Dundee, Dundee, 1973), pp. 12–20. Lecture Notes in Math., Vol. 363. Springer, Berlin (1974)
3. Gander, M., Haynes, R.: Domain decomposition approaches for mesh generation via the equidistribution principle. SIAM Journal on Numerical Analysis **50**(4), 2111–2135 (2012)
4. Gander, M.J.: Schwarz methods over the course of time. Electron. Trans. Numer. Anal. **31**, 228–255 (2008)
5. Gander, M.J., Haynes, R.D., Howse, A.J.: Alternating and linearized alternating schwarz methods for equidistributing grids (2012). In Press, 8 pages
6. Haynes, R., Howse, A.: Alternating schwarz methods for mesh equidistribution (Submitted Oct 5, 2012)
7. Huang, W., Russell, R.D.: Adaptive moving mesh methods, *Applied Mathematical Sciences*, vol. 174. Springer, New York (2011)
8. Huang, W.Z., Sloan, D.M.: A simple adaptive grid method in two dimensions. SIAM J. Sci. Comput. **15**(4), 776–797 (1994)
9. Mackenzie, J.A.: The efficient generation of simple two-dimensional adaptive grids. SIAM J. Sci. Comput. **19**(4), 1340–1365 (electronic) (1998)

MPI–OpenMP algorithms for the parallel space–time solution of Time Dependent PDEs

Ronald D. Haynes[1] and Benjamin W. Ong[2]

1 Introduction

Modern high performance computers offer hundreds of thousands of processors that can be leveraged, in parallel, to compute numerical solutions to time dependent partial differential equations (PDEs). For grid-based solutions to these PDEs, domain decomposition (DD) is often employed to add spatial parallelism [19].

Parallelism in the time variable is more difficult to exploit due to the inherent causality. Recently, researchers have explored this issue as a means to improve the scalability of existing parallel spatial solvers applied to time dependent problems. There are several general approaches to combine temporal parallelism with spatial parallelism. Waveform relaxation [15] is an example of a "parallel across the problem" method. The "parallel across the time domain" approaches include the parareal method [11, 17, 16]. The parareal method decomposes a time domain into smaller temporal subdomains and alternates between applying a coarse (relatively fast) sequential solver to compute an approximate (not very accurate) solution, and applying a fine (expensive) solver on each temporal subdomain in parallel. Alternatively, one can consider "parallel across the step" methods. Examples of such approaches include the computation of intermediate Runge–Kutta stage values in parallel [18], and Revisionist Integral Deferred Correction (RIDC) methods, which are the family of parallel time integrators considered in this paper. Parallel across the step methods allow for "small scale" parallelism in time. Specifically, we will show that if a DD implementation scales to N_x processors, a RIDC-DD parallelism will scale to $N_t \times N_x$ processors, where $N_t < 12$ in practice. This contrasts with parallel across the time domain approaches, which can potentially utilize $N_t \gg 12$.

This paper discusses the implementation details and profiling results of the parallel space–time RIDC-DD algorithm described in [5]. Two hybrid OpenMP – MPI frameworks are discussed: (i) a more traditional fork-join approach of combining threads before doing MPI communications, and (ii) a threaded MPI communications framework. The latter framework is highly desirable because existing (spatially parallel) legacy software can be easily integrated with the parallel time integrator. Numerical experiments measure the communication overhead of both frameworks, and demonstrate that the fork-join approach scales well in space and time. Our results indicate that one should strongly consider temporal parallelization for the solution of time dependent PDEs.

[1] Memorial University of Newfoundland, St. John's, Newfoundland, Canada e-mail: rhaynes@mun.ca .[2] Michigan State University, Institute for Cyber-Enabled Research, East Lansing, MI, USA e-mail: ongbw@msu.edu

2 Review

This paper is interested in parallel space-time solutions to the linear heat equation. We describe the application of our method to the linear heat equation in one spatial dimension $x \in [0,1]$ and $t \in [0,T]$,

$$u_t = u_{xx},\, u(t,0) = g_0(t),\, u(t,1) = g_1(t),\, u(0,x) = u_0(x). \tag{1}$$

The actual numerical results in §4 are presented for the 2D heat equation.

2.1 RIDC

RIDC methods [6, 7] are a family of parallel time integrators that can be broadly classified as predictor corrector algorithms [10, 2]. The basic idea is to simultaneously compute solutions to the PDE of interest and associated error PDEs using a low-order time integrator. We first review the derivation of the error equation.

Suppose $v(t,x)$ is an approximate solution to (1), and $u(t,x)$ is the (unknown) exact solution. The error in the approximate solution is $e(t,x) = u(t,x) - v(t,x)$. We define the residual as $\varepsilon(t,x) = v_t(t,x) - v_{xx}(t,x)$. Then the time derivative of the error satisfies $e_t = u_t - v_t = u_{xx} - (v_{xx} + \varepsilon)$. The integral form of the error equation,

$$\left[e + \int_0^t \varepsilon(\tau,x)\, d\tau \right]_t = (v+e)_{xx} - v_{xx}, \tag{2}$$

can then be solved for $e(t,x)$ using the initial condition $e(0,x) = 0$. The correction $e(t,x)$ is combined with the approximate solution $v(t,x)$ to form an improved solution. This improved solution can be fed back in to the error equation (2) and the process repeated until a sufficiently accurate solution is obtained. It has been shown that each application of the error equation improves the order of the overall method, provided the integral is approximated with sufficient accuracy using quadrature [8].

We introduce some notation to identify the sequence of corrected approximations. Denote $v^{[p]}(t,x)$ as the approximate solution which has error $e^{[p]}(t,x)$, which is obtained by solving

$$\left[e^{[p]} + \int_0^t \varepsilon^{[p]}(\tau,x)\, d\tau \right]_t = (v^{[p]} + e^{[p]})_{xx} - v_{xx}^{[p]}, \tag{3}$$

where $v^{[0]}(t,x)$ denotes the initial approximate solution obtained by solving the physical PDE (1) using a low-order integrator. In general, the error from the pth correction equation is used to construct the $(p+1)st$ approximation, $v^{[p+1]}(t,x) = v^{[p]}(t,x) + e^{[p]}(t,x)$. Hence, equation (3) can be expressed as

$$\left[v^{[p+1]} - \int_0^t v_{xx}^{[p]}(\tau,x)\, d\tau \right]_t = v_{xx}^{[p+1]} - v_{xx}^{[p]}. \tag{4}$$

We compute a low-order prediction, $v^{[0],n+1}$, for the solution of (1) at time t_{n+1} using a first-order backward Euler discretization (in time):

$$v^{[0],n+1} - \Delta t\, v_{xx}^{[0],n+1} = v^{[0],n},\ v^{[0],n+1}(a) = g_0(t^{n+1}),\ v^{[0],n+1}(b) = g_1(t^{n+1}), \quad (5)$$

with $v^{[0],0}(x) = u_0(x)$. With some algebra, a first-order backward Euler discretization of equation (4) gives the update, $v^{[p+1],n+1}$, as

$$v^{[p+1],n+1} - \Delta t\, v_{xx}^{[p+1],n+1} = v^{[p+1],n} - \Delta t\, v_{xx}^{[p],n+1} + \int_{t^n}^{t_{n+1}} v_{xx}^{[p]}(\tau, x)\, d\tau, \quad (6)$$

with $v^{[p+1],n+1}(a) = g_0(t^{n+1})$ and $v^{[p+1],n+1}(b) = g_1(t^{n+1})$. The integral in equation (6) is approximated using a sufficiently high-order quadrature rule [8].

Parallelism in time is possible because the PDE of interest (1) and the error PDEs (4) can be solved simultaneously, after initial startup costs. The idea is to fill out the memory footprint, which is needed so that the integral in equation (6) can be approximated by high-order quadrature, before marching solutions to (5) and (6) in a pipe–line fashion. See Figure 1 for a graphical example, and [6] for more details.

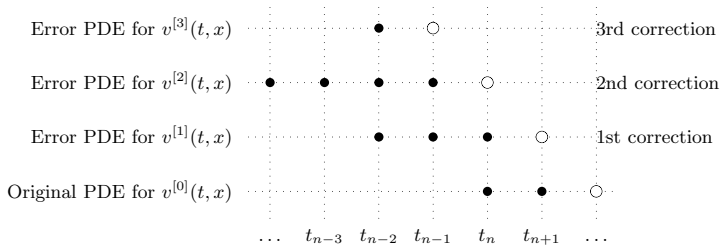


Fig. 1 The black dots represent the memory footprint that must be stored before the white dots can be computed in a pipe. In this figure, $v^{[0],n+2}(x)$, $v^{[1],n+1}(x)$, $v^{[2],n}(x)$ and $v^{[3],n-1}(x)$ are computed simultaneously.

2.2 RIDC–DD

The RIDC–DD algorithm solves the predictor (5) and corrections (6) using DD algorithms in space. The key observation is that (5) and (6) are **both** elliptic PDEs of the form $(1 - \Delta t\, \partial_{xx})z = f(x)$. The function $f(x)$ is known from the solution at the previous time step and previously computed lower-order approximations. DD algorithms for solving elliptic PDEs are well known [3, 4]. The general idea is to replace the PDE by a coupled system of PDEs over some partitioning of the spatial domain using overlapping or non–overlapping subdomains. The coupling is provided by necessary transmission conditions at the subdomain boundaries. These transmission conditions are chosen to ensure the DD algorithm converges and to optimize the con-

vergence rate. In [5], as a proof of principle, (5-6) are solved using a classical parallel Schwarz algorithm, with overlapping subdomains and Dirichlet transmission conditions. Optimized RIDC–DD variants are possible using an optimized Schwarz DD method [13, 12, 9], to solve (5-6). The solution from the previous time step can be used as initial subdomain solutions at the interfaces. We will use RIDCp–DD to refer to a pth-order solution obtained using $p-1$ RIDC corrections in time and DD in space.

3 Implementation Details

We view the parallel time integrator reviewed in §2.1 as a simple yet powerful tool to add further scalability to a legacy MPI or modern MPI–CUDA code, while improving the accuracy of numerical solution. The RIDC integrators benefit from access to shared memory because solving the correction PDE (6) requires both the solution from the previous time step and previously computed lower-order subdomain solution. Consequently, we propose two MPI-OpenMP hybrid implementations which map well to multi-core, multi-node compute resources. In the upcoming MPI 3.0 standard [1], shared memory access within the MPI library will provide alternative implementations.

Implementation #1: The RIDC-DD algorithm can be implemented using a traditional fork join approach, as illustrated in Program 1. After boundary information is exchanged, each MPI task spawns OpenMP threads to perform the linear solve. The threads are merged back together before MPI communication is used to check for convergence. The drawback to this fork-join implementation, is that the parallel space-time algorithm becomes tightly integrated, making it difficult to leverage an existing spatially parallel DD implementation.

```
1.  MPI Initialization
2.  ...
3.    for each time step
4.       for each Schwarz iteration
5.          MPI Comm (exchange boundary info)
6.             OMP Parallel for each prediction/correction
7.                linear solve
8.             end parallel
9.          MPI Comm (check for convergence)
10.      end
11.   end
12. ...
13. MPI Finalize
```

Program 1: RIDC-DD implementation using a fork-join approach. The time parallelism occurs *within* each Schwarz iteration, requiring a tight integration with an existing spatially parallel DD implementation.

Implementation #2: To leverage an existing spatially parallel DD implementation, a non-traditional hybrid approach must be considered. By changing the order of the loops, the Schwarz iterations for the prediction and the correction loops can be evaluated independently of each other. This is realized by spawning individual OpenMP threads to solve the prediction and correction loops on each subdomain; the Schwarz iterations for the prediction/correction step run independently of each other until convergence. This implementation (Program 2) has several consequences: (i) a thread safe version of MPI supporting `MPI_THREAD_MULTIPLE` is required. (ii) In addition, we required a thread-safe, thread-independent version of `MPI_BARRIER`, `MPI_BROADCAST` and `MPI_GATHER`. To achieve this, we wrote our own wrapper library using the thread safe `MPI_SEND`, `MPI_RECV` and `MPI_SENDRECV` provided by (i).

```
1.  MPI Initialization
2.  ...
3.    for each time step
4.       OMP Parallel for each prediction/correction level
5.          for each Schwarz iteration
6.             MPI Comm (exchange boundary info)
7.             linear solve
8.             MPI Comm (check for convergence)
9.          end
10.      end parallel
11.   end
12. ...
13. MPI Finalize
```

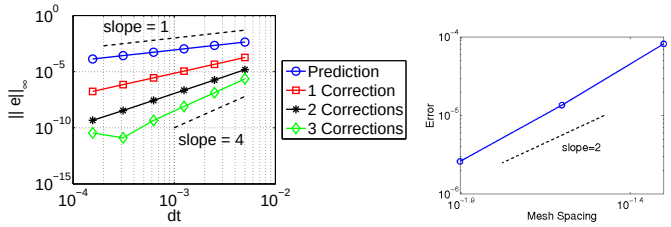
Program 2: RIDC-DD implementation using a non-traditional hybrid approach. Notice that lines 5-9 are the Schwarz iterations that one would find in an existing spatially parallel DD implementation. Hence, provided the DD implementation is thread-safe, one could wrap the time parallelization around an existing parallel DD implementation.

4 Numerical Experiments

We show first that RIDC-DD methods converge with the designed orders in space and time. Then, we profile communication costs using TAU [14]. Finally, we show strong scaling studies for the RIDC-DD algorithm. We compute solutions to the heat equation in $\mathbb{R}^2$, where centered finite differences are used to approximate the second derivative operator. Errors are computed using the known analytic solution. The computations are performed at the High Performance Computing Cenrer at Michigan State University, where nodes (consisting of two quad core Intel Westmere processors) are interconnected using infiniband and a high speed Lustre file system.

4.1 Convergence Studies and Profile Analysis

In Figure 2, the convergence plots show that our classical Schwarz RIDC-DD algorithm converges as expected in space and time. In general, one would balance the orders of the errors in space and time appropriately for efficiency. Here we pick RIDC4 since it mapped well to our available four core sockets and to demonstrate the scalability of our algorithm in time. We could, of course, used a fourth order method in space. The Schwarz iterations are iterated until a tolerance of 10^{-12} is reached for the predictors and correctors (which explains why the error in the fourth-order approximation levels out as the time step becomes small).



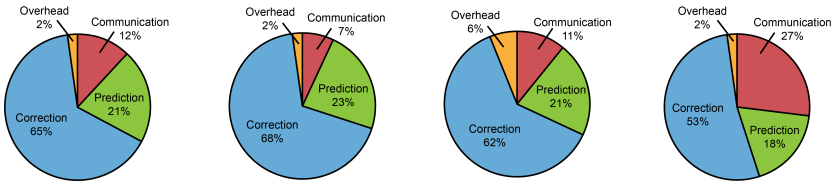
(a) Time Convergence (b) Space Convergence

Fig. 2 (a) Classical Schwarz RIDCp-DD algorithms, $p = 1,2,3,4$, converge to the reference solution with the designed orders of accuracy. Here Δx is fixed while Δt is varied. (b) Second-order convergence in space is demonstrated for the fourth-order RIDC-DD algorithm. Here, Δt is fixed while Δx is varied.

The communication costs for our two implementations of RIDC4-DD are profiled using TAU [14]. We see in Figure 3, communication costs are minimal for implementation #1, and scales nicely as the number of nodes is increased, but the communication cost is significant for implementation #2. In Figure 3(a,c), the domain is discretized into 180×180 grid nodes, which are split into a 3×3 configuration of subdomains. In Figure 3(b,d), the domain is discretized into 360×360 grid nodes, which are split into a 6×6 configuration of subdomains. This keeps the number of grid points per subdomain constant so that the computation time for the matrix factorization and linear solve are the same.

4.2 Characterizing Parallel Performance

Due to the better communication profile, we use framework #1 for our experiments. We fix $\Delta x = \frac{1}{180}$, $\Delta y = \frac{1}{180}$, $\Delta t = \frac{1}{1000}$, and TOL=$10^{-12}$ (the Schwarz iteration tolerance). We consider three configurations of subdomains: 2×2, 4×4 and 6×6. For each configuration we illustrate the speedup and efficiency due to the time parallelism in Figure 4. We choose to fix the ratio between the overlap and subdomain



(a) Implementation #1 3×3 domain (b) Implementation #1 6×6 domain (c) Implementation #2 3×3 domain (d) Implementation #2 6×6 domain

Fig. 3 Profile of the RIDC4-DD algorithm using both implementations. Overhead and communication costs are reasonable for implementation #1, but are high for implementation #2.

size to ensure the number of unknowns on each subdomain scales appropriately as the number of subdomains is increased.

In Figure 4 we show three curves corresponding to a 2×2, 4×4 and a 6×6 configuration of subdomains. For each configuration we compute a fourth order solution in time using $1, 2$ and 4 threads. The 6×6 configuration of subdomains with 4 threads uses a total of 144 cores. We plot the efficiency (with respect to the one thread run) as a function of the number of threads. Speedup is evident as temporal parallelization is improved, however, efficiency decreases as the number of subdomains increases.

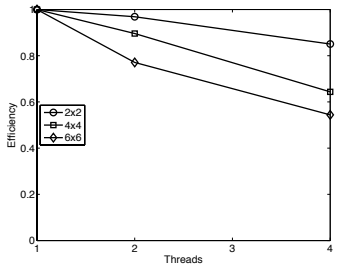


Fig. 4 Scaling study (in time) for a RIDC4-DD algorithm.

5 Conclusions

This paper has presented the implementation details and first reported profiling results for a newly proposed space–time parallel algorithm for time dependent PDEs. The RIDC–DD method combines traditional domain decomposition in space with a new family of deferred correction methods designed to allow parallelism in time. Two possible implementations are described and profiled. The first, a traditional hybrid OpenMP–MPI implementation, requires potentially difficult modifications of an existing parallel spatial solver. Numerical experiments verify that the algorithm achieves its designed order of accuracy and scales well. The second strategy allows a relatively easy reuse of an existing parallel spatial solver by using OpenMP to spawn threads for the simultaneous prediction and correction steps. This non–traditional hybrid use of OpenMP and MPI currently requires writing of custom thread–safe and thread–independent MPI routines. Profile analysis shows that our non-traditional use of OpenMP–MPI suffers from higher communication costs than the standard use of OpenMP-MPI. An inspection of the prediction and correction

equations indicates that optimized Schwarz variants of the algorithm are possible and will enjoy nice load balancing. This work is ongoing.

Acknowledgements This work was supported by the Institute for Cyber-Enabled Research (iCER) at MSU, NSERC Discovery Grant 311796, and AFOSR Grant FA9550-12-1-0455.

References

1. Mpi 3.0 standardization effort. `http://meetings.mpi-forum.org/MPI_3.0_main_page.php`. Accessed 10/25/2012
2. Böhmer, K., Stetter, H.: Defect correction methods. Theory and applications. Computing Supplementum, 5 (1984)
3. Cai, X.C.: Additive Schwarz algorithms for parabolic convection-diffusion equations. Numer. Math. **60**(1), 41–61 (1991)
4. Cai, X.C.: Multiplicative Schwarz methods for parabolic problems. SIAM J. Sci. Comput. **15**(3), 587–603 (1994)
5. Christlieb, A., Haynes, R., Ong, B.: A parallel space-time algorithm. SIAM J. Sci. Comput. **34**(5), 233–248 (2012)
6. Christlieb, A., Macdonald, C., Ong, B.: Parallel high-order integrators. SIAM J. Sci. Comput. **32**(2), 818–835 (2010)
7. Christlieb, A., Ong, B.: Implicit parallel time integrators. J. Sci. Comput. **49**(2), 167–179 (2011)
8. Christlieb, A., Ong, B., Qiu, J.M.: Comments on high order integrators embedded within integral deferred correction methods. Comm. Appl. Math. Comput. Sci. **4**(1), 27–56 (2009)
9. Dubois, O., Gander, M., Loisel, S., St-Cyr, A., Szyld, D.: The optimized Schwarz method with a coarse grid correction. SIAM J. Sci. Comput. **34**(1), A421–A458 (2012)
10. Dutt, A., Greengard, L., Rokhlin, V.: Spectral deferred correction methods for ordinary differential equations. BIT **40**(2), 241–266 (2000)
11. Gander, M., Vandewalle, S.: On the superlinear and linear convergence of the parareal algorithm. Lecture Notes in Computational Science and Engineering **55**, 291 (2007)
12. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**(2), 699–731 (2006)
13. Gander, M.J., Halpern, L.: Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems. SIAM J. Numer. Anal. **45**(2), 666–697 (2007)
14. Koehler, S., Curreri, J., George, A.: Performance analysis challenges and framework for high-performance reconfigurable computing. Parallel Computing **34**(4), 217–230 (2008)
15. Lelarasmee, E., Ruehli, A.E., Sangiovanni-Vincentelli, A.L.: The waveform relaxation method for time–domain analysis of large scale integrated circuits. IEEE Trans. on CAD of IC and Syst. **1**, 131–145 (1982)
16. Lions, J., Maday, Y., Turinici, G.: A "parareal" in time discretization of PDEs. Comptes Rendus de l'Academie des Sciences Series I Mathematics **332**(7), 661–668 (2001)
17. Minion, M., Williams, S.: Parareal and spectral deferred corrections. In: NUMERICAL ANALYSIS AND APPLIED MATHEMATICS: International Conference on Numerical Analysis and Applied Mathematics 2008. AIP Conference Proceedings, vol. 1048, pp. 388–391 (2008)
18. Nievergelt, J.: Parallel methods for integrating ordinary differential equations. Communications of the ACM **7**(12), 731–733 (1964)
19. Toselli, A., Widlund, O.: Domain decomposition methods—algorithms and theory, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin (2005)

Neumann–Neumann Waveform Relaxation for the Time-Dependent Heat Equation

Felix Kwok[1]

1 Introduction

The goal of this paper is to introduce and analyze a new variant of waveform relaxation (WR) methods based on Neumann–Neumann iterations. Originally introduced by [13] for ODE systems, WR methods have first been used to solve time-dependent PDEs in [11] and [12]. When applying a WR method for a given domain Ω and a decomposition into subdomains $\{\Omega_i\}_{i=1}^N$, $\cup_i \overline{\Omega_i} = \overline{\Omega}$, each iteration consists of solving independent subproblems on $\Omega_i \times [0,T]$, i.e., over the *whole time window* $[0,T]$, before exchanging information across the interfaces; in other words, the information exchanged consists of interface traces over the time window $[0,T]$. This is in contrast with the classical approach, in which one fixes a time stepping strategy for the whole domain Ω and uses domain decomposition to solve the resulting spatial problem at each time step. One advantage of the WR framework is that it allows the use of different spatial and time discretizations for each subdomain; this is especially useful for problems with large coefficient jumps [9] or with different models for different parts of the domain [8]. In addition, since communication between subdomains are less frequent than for the standard approach, there is a reduction in communication costs, particularly for networks with high latency.

Typically, WR methods can be derived from methods for elliptic PDEs. For example, one can extend the parallel Schwarz method with classical transmission conditions [14] to obtain the parallel Schwarz WR method; this has been analyzed in [11, 12]. WR extensions based on optimized Schwarz methods [6] have also been proposed. Substructuring methods form another class of methods for elliptic PDEs: examples include the Neumann–Neumann method [2, 4], as well as the FETI method [5] and its variants. However, to the best of our knowledge, no substructuring-type analogue of WR has been proposed, despite substructuring methods having many attractive properties for elliptic problems, such as mesh independence in the two-subdomain case. Thus, our first aim is to define the Neumann–Neumann waveform relaxation (NNWR) method, which generalizes the elliptic Neumann–Neumann method in a natural way. This is done in Section 2.

Our second goal is to understand the convergence of the proposed algorithm for parabolic problems. For systems of ODEs, a Picard–Lindelöf type argument shows that convergence is superlinear on bounded time intervals $[0,T]$, with an error estimate of the form $(CT)^k/k!$ after k iterations [16]. For overlapping Schwarz WR methods applied to the advection-diffusion equation, the estimate can be improved

[1] Université de Genève, 2-4 Rue du Lièvre, 1211 Genève, Switzerland, e-mail: felix.kwok@unige.ch

to $e^{-(kL)^2/T}$, where L is the size of the overlap [12]; this bound is possible because of the diffusivity of the problem. However, for unbounded time intervals, only linear convergence can be expected [11]. Similar conclusions hold for Schwarz WR with optimized transmission conditions, with or without overlap [15, 7, 1]. Using the 1D heat equation as the model problem, we show that the NNWR method also converges superlinearly for finite time intervals; this is done in Section 3, with some numerical experiments confirming the results in Section 4. We also derive a linear bound that is valid for unbounded time intervals. We have chosen to analyze the method in the continuous setting because it allows us to understand the asymptotic behaviour of the algorithm for very fine grids, without requiring explicit knowledge of how each subdomain problem is discretized. For ease of presentation, we prove our results for two subdomains in one spatial dimension; [10] contains further results, some of which are mentioned at the end of Section 4.

2 The NNWR algorithm

Suppose we want to solve the 1D heat equation

$$\partial_t u - \partial_x^2 u = f, \qquad x \in \Omega = (-b,a), \quad t \in (0,T],$$

with initial conditions $u(x,0) = v(x)$ and Dirichlet boundary conditions $u(-b,t) = u_L(t)$, $u(a,t) = u_R(t)$. We consider a decomposition into two non-overlapping subdomains $\Omega_1 = (-b,0)$ and $\Omega_2 = (0,a)$. On the interface $\Gamma = \{0\}$, we are given the initial guess $g^0(t)$, $t \in [0,T]$. Then the NNWR algorithm is given by the following iteration: for $k = 1,2,\ldots$, do

(i) Dirichlet step:

$$\begin{cases} \partial_t u_1^k - \partial_x^2 u_1^k = f(x,t) & \text{on } (-b,0), \\ u_1^k(-b,t) = u_L(t), \\ u_1^k(0,t) = g^{k-1}(t), \\ u_1^k(x,0) = v(x) & \text{on } (-b,0), \end{cases} \qquad \begin{cases} \partial_t u_2^k - \partial_x^2 u_2^k = f(x,t) & \text{on } (0,a), \\ u_2^k(0,t) = g^{k-1}(t), \\ u_2^k(a,t) = u_R(t), \\ u_2^k(x,0) = v(x) & \text{on } (0,a). \end{cases}$$

(ii) Neumann step:

$$\begin{cases} \partial_t \psi_1^k - \partial_x^2 \psi_1^k = 0 & \text{on } (-b,0), \\ \psi_1^k(-b,t) = 0, \\ \partial_{n_1} \psi_1^k = \partial_{n_1} u_1^k + \partial_{n_2} u_2^k & \text{on } \Gamma, \\ \psi_1^k(x,0) = 0 & \text{on } (-b,0), \end{cases} \qquad \begin{cases} \partial_t \psi_2^k - \partial_x^2 \psi_2^k = 0 & \text{on } (0,a), \\ \partial_{n_2} \psi_2^k = \partial_{n_1} u_1^k + \partial_{n_2} u_2^k & \text{on } \Gamma, \\ \psi_2^k(a,t) = 0, \\ \psi_2^k(x,0) = 0, & \text{on } (0,a). \end{cases}$$

(iii) Update step:

$$g^k(t) = g^{k-1}(t) - \theta[\psi_1^k(0,t) + \psi_2^k(0,t)].$$

The relaxation parameter $\theta \in (0,1]$ is chosen to obtain fast convergence. Note that this algorithm can be generalized in a straightforward way to handle decompositions into many subdomains and in higher dimensions, see [10]. This is because, unlike for the elliptic case, the Neumann step is always well-posed for the heat equation, even for "floating" subdomains that do not share an edge with $\partial\Omega$.

Analysis by Laplace transforms. Our convergence analysis is based on the Laplace transform method. The Laplace transform of a function $u(x,t)$ with respect to time is defined as

$$\hat{u}(x,s) := \mathscr{L}\{u(x,t)\} = \int_0^\infty u(x,t)e^{-st}\,dt.$$

In the remainder of the paper, hats will denote the Laplace transform of a function in time, and s will denote the Laplace variable. Since we are interested in the error $g^k(t) - u(0,t)$ of the method, it suffices to assume that $v(x)$, $f(x,t), u_L(t)$ and $u_R(t)$ all vanish and study how $g^k(t)$ tends to zero as $k \to \infty$. In this case, the NNWR algorithm can be written in Laplace space as follows: for $k = 1,2,\ldots$, do

(i) Dirichlet step:

$$\begin{cases} (s-\partial_x^2)\hat{u}_1^k = 0 & \text{on } (-b,0), \\ \hat{u}_1^k(-b,s) = 0, \\ \hat{u}_1^k(0,s) = \hat{g}^{k-1}(s), \end{cases} \qquad \begin{cases} (s-\partial_x^2)\hat{u}_2^k = 0 & \text{on } (0,a), \\ \hat{u}_2^k(0,s) = \hat{g}^{k-1}(s), \\ \hat{u}_2^k(a,t) = 0. \end{cases}$$

(ii) Neumann step:

$$\begin{cases} (s-\partial_x^2)\hat{\psi}_1^k = 0 & \text{on } (-b,0), \\ \hat{\psi}_1^k(-b,s) = 0, \\ \partial_x\hat{\psi}_1^k = \partial_x\hat{u}_1^k - \partial_x\hat{u}_2^k & \text{on } \Gamma, \end{cases} \qquad \begin{cases} (s-\partial_x^2)\hat{\psi}_2^k = 0 & \text{on } (0,a), \\ -\partial_x\hat{\psi}_2^k = \partial_x\hat{u}_1^k - \partial_x\hat{u}_2^k & \text{on } \Gamma, \\ \hat{\psi}_2^k(a,s) = 0. \end{cases}$$

(iii) Update step:

$$\hat{g}^k(s) = \hat{g}^{k-1}(s) - \theta[\hat{\psi}_1^k(0,s) + \hat{\psi}_2^k(0,s)].$$

Solving the two-point boundary value problems in the Dirichlet step yields

$$\hat{u}_1^k(x,s) = \hat{g}^{k-1}(s)\frac{\sinh((x+b)\sqrt{s})}{\sinh(b\sqrt{s})}, \qquad \hat{u}_2^k(x,s) = \hat{g}^{k-1}(s)\frac{\sinh((a-x)\sqrt{s})}{\sinh(a\sqrt{s})}. \quad (1)$$

The Neumann step can be solved similarly by letting $\hat{r}^k(s) := \partial_x\hat{u}_1^k(0,s) - \partial_x\hat{u}_2^k(0,s)$:

$$\hat{\psi}_1^k(x,s) = \hat{r}^k(s)\frac{\sinh((x+b)\sqrt{s})}{\sqrt{s}\cosh(b\sqrt{s})}, \qquad \hat{\psi}_2^k(x,s) = \hat{r}^k(s)\frac{\sinh((a-x)\sqrt{s})}{\sqrt{s}\cosh(a\sqrt{s})}. \quad (2)$$

Then the update step becomes

$$\hat{g}^k(s) = \hat{g}^{k-1}(s) - \theta[\psi_1^k(0,s) + \psi_2^k(0,s)] = \hat{g}^{k-1}(s) - \theta\frac{\hat{r}^k(s)}{\sqrt{s}}[\tanh(b\sqrt{s}) + \tanh(a\sqrt{s})].$$

But

$$\hat{r}^k(s) = \partial_x u_1^k(0,s) - \partial_x u_2^k(0,s) = \sqrt{s}\hat{g}^{k-1}(s)\left(\frac{\cosh(b\sqrt{s})}{\sinh(b\sqrt{s})} + \frac{\cosh(a\sqrt{s})}{\sinh(a\sqrt{s})}\right).$$

So we obtain

$$\hat{g}^k(s) = \hat{g}^{k-1}(s)\left[1 - \theta\left(2 + \frac{\tanh(a\sqrt{s})}{\tanh(b\sqrt{s})} + \frac{\tanh(b\sqrt{s})}{\tanh(a\sqrt{s})}\right)\right]. \tag{3}$$

Note that if $a = b$, then $\hat{g}^k(s) = \hat{g}^{k-1}(s)(1-4\theta)$, which means *the method converges to the exact solution in one iteration for* $\theta = 1/4$. Thus, the classical result for elliptic problems also holds for the time-dependent case. The main result of our paper concerns the case when the subdomains are unequal, i.e., when $a \neq b$.

Theorem 1 (Convergence of NNWR). *Let* $\theta = 1/4$. *Then the error of the NNWR method for two subdomains satisfies*

$$\|g^k(\cdot) - u(0,\cdot)\|_{L^\infty(0,\infty)} \leq \left(\frac{(a-b)^2}{4ab}\right)^k \|g^0(\cdot) - u(0,\cdot)\|_{L^\infty(0,\infty)}. \tag{4}$$

Moreover, for every finite time interval $(0,T)$, *NNWR converges superlinearly with the estimate*

$$\|g^k(\cdot) - u(0,\cdot)\|_{L^\infty(0,T)} \leq e^{-k^2m^2/T}\left(\frac{(a-b)^2}{ab}\right)^k \|g^0(\cdot) - u(0,\cdot)\|_{L^\infty(0,T)}, \tag{5}$$

where $m = \min\{a,b\}$.

3 Convergence analysis

Since (3) is symmetric with respect to a and b, we will assume without loss of generality that $a > b$ in the remainder of the paper. For $\theta = 1/4$, the recurrence (3) simplifies to give

$$\hat{g}^k(s) = -\hat{g}^{k-1}(s) \cdot \frac{\sinh^2((a-b)\sqrt{s})}{\sinh(2a\sqrt{s})\sinh(2b\sqrt{s})} =: -Y(s)\hat{g}^{k-1}(s), \tag{6}$$

which implies $\hat{g}^k(s) = (-1)^k Y^k(s)\hat{g}^0(s)$. Note that for $\mathrm{Re}(s) > 0$, we have $Y(s) = O(e^{-4b|s|^{1/2}})$ as $|s| \to \infty$, i.e., $Y(s)$ decays exponentially as $|s| \to \infty$. Thus, by [3, p. 183], $Y(s)$ is the Laplace transform of a regular function $y_1(t)$. If we now define $y_k(t) = \mathscr{L}^{-1}\{Y^k(s)\}$, then for $t \in (0,T)$, we have

$$|g^k(t)| = \left|\int_0^t g^0(t-\tau)y_k(\tau)\,d\tau\right| \leq \|g^0\|_{L^\infty(0,T)} \int_0^T |y_k(\tau)|\,d\tau. \tag{7}$$

Thus, to obtain L^∞ convergence estimates, we need bounds on $\int_0^T |y_k(\tau)|\,d\tau$. Our first step is to show that $y_k(t) \geq 0$, for $t > 0$, which makes bounding its integral much easier. We start by stating a few elementary properties of positive functions and their Laplace transforms; their proofs follow easily from the definitions.

Lemma 1. *Let f and g be positive functions, i.e., $f(t) \geq 0$ and $g(t) \geq 0$ for $t > 0$, and let $F(s) = \mathscr{L}\{f(t)\}$. Then*

(i) For all $T > 0$, $\displaystyle\int_0^T |f(\tau)|\,d\tau \leq \int_0^\infty f(\tau)\,d\tau = \lim_{s\to 0} F(s).$

(ii) $\displaystyle(f*g)(t) = \int_0^t f(t-\tau)g(\tau)\,d\tau \geq 0$ *for all $t > 0$.*

(iii) $\|f*g\|_{L^1(0,T)} \leq \|f\|_{L^1(0,T)} \cdot \|g\|_{L^1(0,T)}$.

Lemma 2. *For $\beta > \alpha \geq 0$, let*

$$Q_1(s) = \frac{\sinh(\alpha\sqrt{s})}{\sinh(\beta\sqrt{s})}, \qquad Q_2(s) = \frac{\cosh(\alpha\sqrt{s})}{\cosh(\beta\sqrt{s})}.$$

Then $q_1(t) = \mathscr{L}^{-1}\{Q_1(s)\}$ and $q_2(t) = \mathscr{L}^{-1}\{Q_2(s)\}$ are positive functions.

Proof. For $n = 1, 2, \ldots$, let $u_n(x,t)$ and $v_n(x,t)$ be the solutions of the following two boundary value problems:

$$\begin{cases} \partial_t u_n - \partial_x^2 u_n = 0 & \text{on } (0,\beta), \\ u_n(0,t) = 0, \\ u_n(\beta,t) = f_n(t), \\ u_n(x,0) = 0, \end{cases} \qquad \begin{cases} \partial_t w_n - \partial_x^2 w_n = 0 & \text{on } (-\beta,\beta), \\ w_n(-\beta,t) = f_n(t), \\ w_n(\beta,t) = f_n(t), \\ w_n(x,0) = 0. \end{cases}$$

A calculation similar to that in Section 2 shows that $\mathscr{L}\{u_n(\alpha,t)\} = Q_1(s)\hat{f}_n(s)$ and $\mathscr{L}\{w_n(\alpha,t)\} = Q_2(s)\hat{f}_n(s)$. Moreover, if $f_n(t) \geq 0$ for all t, then by the maximum principle, we have $u_n(\alpha,t) \geq 0$. We now choose a sequence (f_n) of positive functions that converges weakly to $\delta(t)$; then since each $u_n(\alpha,t)$ is positive, we have $u_n(\alpha,t) \to q_1(t) \geq 0$. A similar argument shows that $w_n(\alpha,t) \to q_2(t) \geq 0$. □

We now analyze the kernel $y_1(t)$, with Laplace transform $Y(s)$, as defined in (6).

Lemma 3. *Let $m \geq 1$ be the unique integer such that $mb < a \leq (m+1)b$. Then $Y(s) = V(s)H(s)$, with $V(s) = 1/\cosh^2(b\sqrt{s})$ and $\lim_{s\to 0} H(s) = (a-b)^2/4ab$. Moreover, $h(t) = \mathscr{L}^{-1}\{H(s)\}$ is positive, so that $y_1(t) = (v*h)(t) \geq 0$ for all $t > 0$.*

Proof. For $k < m$, we have the identity

$$\begin{aligned} &\sinh^2((a-kb)\sqrt{s}) - \sinh^2((a-(k+1)b)\sqrt{s}) \\ &\qquad = \frac{1}{2}\left[\cosh(2(a-kb)\sqrt{s}) - 1 - \cosh(2(a-(k+1)b)\sqrt{s}) + 1\right] \\ &\qquad = \sinh((2a-(2k+1)b)\sqrt{s})\sinh(b\sqrt{s}). \end{aligned}$$

Since $k < m$, we have $0 < 2a-(2k+1)b < 2a$, which gives

$$\frac{\sinh^2((a-kb)\sqrt{s})}{\sinh(2a\sqrt{s})\sinh(2b\sqrt{s})} = \frac{\sinh((2a-(2k+1)b)\sqrt{s})}{\sinh(2a\sqrt{s})}\cdot\frac{\sinh(b\sqrt{s})}{\sinh(2b\sqrt{s})} + \frac{\sinh^2((a-(k+1)b)\sqrt{s})}{\sinh(2a\sqrt{s})\sinh(2b\sqrt{s})}.$$

Applying this identity repeatedly for $k = 1,\dots,m-1$ gives

$$\begin{aligned}
Y(s) &= \frac{\sinh^2((a-b)\sqrt{s})}{\sinh(2a\sqrt{s})\sinh(2b\sqrt{s})} \\
&= \frac{\sinh^2((a-mb)\sqrt{s})}{\sinh(2a\sqrt{s})\sinh(2b\sqrt{s})} + \sum_{k=1}^{m-1}\frac{\sinh((2a-(2k+1)b)\sqrt{s})}{\sinh(2a\sqrt{s})}\cdot\frac{\sinh(b\sqrt{s})}{\sinh(2b\sqrt{s})} \\
&= \frac{1}{2\cosh^2(b\sqrt{s})}\left[\frac{\sinh^2((a-mb)\sqrt{s})\cosh(b\sqrt{s})}{\sinh(2a\sqrt{s})\sinh(b\sqrt{s})} + \sum_{k=1}^{m-1}\frac{\sinh((2a-(2k+1)b)\sqrt{s})\cosh(b\sqrt{s})}{\sinh(2a\sqrt{s})}\right] \\
&= \frac{1}{4\cosh^2(b\sqrt{s})}\left[\frac{\sinh((a-mb)\sqrt{s})}{\sinh(a\sqrt{s})}\cdot\frac{\sinh((a-mb)\sqrt{s})}{\sinh(b\sqrt{s})}\cdot\frac{\cosh(b\sqrt{s})}{\cosh(a\sqrt{s})} + \right. \\
&\qquad \left. \sum_{k=1}^{m-1}\left(\frac{\sinh((2a-2kb)\sqrt{s})}{\sinh(2a\sqrt{s})} + \frac{\sinh((2a-2(k+1)b)\sqrt{s})}{\sinh(2a\sqrt{s})}\right)\right]
\end{aligned}$$

Let $V(s) = 1/\cosh^2(b\sqrt{s})$ and $H(s)$ be the rest. Then since $0 < a-mb \le b < a$, we see that $H(s)$ consists of a sum of products of functions of the form $Q_1(s)$ and $Q_2(s)$ in Lemma 2. Thus, its inverse Laplace transform $h(t)$ is positive. Moreover, since $v(t) = \mathscr{L}^{-1}\{V(s)\}$ is also positive by Lemma 2, we see that $y(t) = (v*h)(t)$ is positive. Finally, since $\lim_{s\to 0}V(s) = 1$, we have

$$\lim_{s\to 0}H(s) = \lim_{s\to 0}Y(s) = \lim_{s\to 0}\frac{\sinh^2((a-b)\sqrt{s})}{\sinh(2a\sqrt{s})\sinh(2b\sqrt{s})} = \frac{(a-b)^2}{4ab}. \qquad \square$$

We are finally ready to prove our main result.

Proof (Theorem 1). According to (7), it suffices to bound $\int_0^T |y_k(\tau)|\,d\tau$ for finite $T > 0$ and for $T = \infty$, where $y_k(t) = \mathscr{L}^{-1}\{Y^k(s)\}$. Since $y_1(t)$ is positive by Lemma 3, so is $y_k(t)$, so by Lemma 1(i), we have

$$\int_0^\infty |y_k(\tau)|\,d\tau = \lim_{s\to 0}Y^k(s) = \left(\frac{(a-b)^2}{4ab}\right)^k,$$

which shows the linear bound (4). For $T < \infty$, let $v_k(t) = \mathscr{L}^{-1}\{V^k(s)\}$ and $h_k(t) = \mathscr{L}^{-1}\{H^k(s)\}$. Then since $\int_0^\infty h_k(t)\,dt = \lim_{s\to 0}H^k(s) = (\lim_{s\to 0}H(s))^k$, we have

$$\|y_k\|_{L^1(0,T)} \le \|v_k\|_{L^1(0,T)}\cdot\|h_k\|_{L^1(0,T)} \le \left(\frac{(a-b)^2}{4ab}\right)^k\int_0^T v_k(\tau)\,d\tau. \tag{8}$$

To bound the remaining integral, let $D(s) = 4^k e^{-2kb\sqrt{s}} - V^k(s)$. We will show that $d(t) = \mathscr{L}^{-1}\{D(s)\} \ge 0$. We have

$$D(s) = 4^k e^{-2kb\sqrt{s}} - \frac{2^{2k}}{(e^{b\sqrt{s}} + e^{-b\sqrt{s}})^{2k}} = 4^k \cdot \frac{(1+e^{-2b\sqrt{s}})^{2k} - 1}{(e^{b\sqrt{s}} + e^{-b\sqrt{s}})^{2k}} = 4^k \sum_{m=1}^{2k} \binom{2k}{m} e^{-2bm\sqrt{s}} V^k(s).$$

From [17], we know that $\mathscr{L}^{-1}\{e^{-2bm\sqrt{s}}\} = \frac{bm}{\sqrt{\pi t^3}} e^{-b^2m^2/t}$ is a positive function for $m \geq 1$. Since $v_k(t) = \mathscr{L}^{-1}\{V^k(s)\}$ is also positive, we see that $d(t)$ is in fact a sum of convolutions of positive functions. Hence $d(t) \geq 0$, as claimed. Thus, we have

$$\int_0^T v_k(\tau)\,d\tau \leq \int_0^T (v_k(\tau) + d(\tau))\,d\tau = \int_0^T 4^k \frac{kb}{\sqrt{\pi\tau^3}} e^{-k^2b^2/\tau}\,d\tau = 4^k \operatorname{erfc}\left(\frac{bk}{\sqrt{T}}\right).$$

But $\operatorname{erfc}(x) \leq e^{-x^2}$ for all $x \geq 0$; introducing this into (8) gives the estimate

$$\|y_k\|_{L^1(0,T)} \leq \left(\frac{(a-b)^2}{ab}\right)^k \operatorname{erfc}\left(\frac{bk}{\sqrt{T}}\right) \leq \left(\frac{(a-b)^2}{ab}\right)^k e^{-k^2b^2/T},$$

which tends to zero as $k \to \infty$.

4 Numerical experiments

Figure 1 shows the convergence of NNWR for a mildly asymmetric case ($a = 0.7$, $b = 0.3$) and a strongly asymmetric case ($a = 0.9$, $b = 0.1$) when applied to a finite-difference Crank–Nicolson discretization. We see that the bounds in Theorem 1, while not necessarily sharp, does capture the superlinear convergence of the method. As the length of the time window T increases, the error curve approaches the linear bound, which can be increasing for highly asymmetric problems. In this case, the error can grow substantially before decreasing to zero superlinearly. Thus, one should divide up the problem into several small time windows before using NNWR.

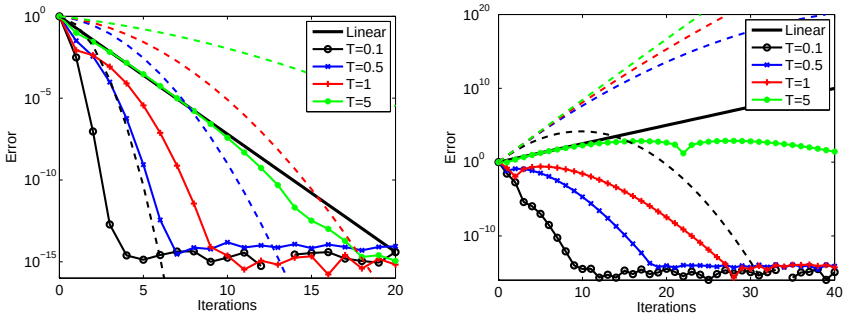


Fig. 1 Convergence curves and their respective bounds for (i) $a = 0.7$, $b = 0.3$ and (ii) $a = 0.9$, $b = 0.1$. The solid curves (with markers) denote the L^∞ error after k iterations for the final time T indicated, and dotted lines of the same color show the superlinear bound (5) for the same T. The linear bound (4) is shown as a solid black line (no markers).

Convergence estimates for more general decompositions can also be obtained. For the 1D heat equation with N subdomains, we have

$$\max_{1\le i\le N} \|e_i^k\|_{L^\infty(0,T)} \le \left(\frac{\sqrt{6}}{1-e^{-(2k+1)/\tau}}\right)^{2k} e^{-k^2/\tau} \max_{1\le i\le N} \|e_i^0\|_{L^\infty(0,T)}, \tag{9}$$

where e_i^k is the error along the ith interface at iteration k and $\tau = T/h^2$, with h being the smallest subdomain size. The estimate (9) is also valid for the 2D heat equation on a rectangular domain decomposed into N strips. For the proofs of these and other results, see [10]. Note that as N increases, the subdomain size h necessarily decreases, and the bound (9) shows that the error can increase before superlinear convergence kicks in, just like in the asymmetric case above. To remedy this, we recommend using a coarse grid correction, which is the subject of a future paper.

References

1. Bennequin, D., Gander, M., Halpern, L.: A homographic best approximation problem with application to optimized Schwarz waveform relaxation. Math. Comp. **78**(265), 185–223 (2009)
2. Bourgat, J.F., Glowinski, R., Le Tallec, P., Vidrascu, M.: Variational formulation and algorithm for trace operator in domain decomposition calculations. In: Second International Symposium on Domain Decomposition Methods, pp. 3–16 (1989)
3. Churchill, R.V.: Operational mathematics, 2nd edn. McGraw-Hill (1958)
4. De Roeck, Y.H., Le Tallec, P.: Analysis and test of a local domain decomposition preconditioner. In: Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, pp. 112–128 (1991)
5. Farhat, C., Roux, F.X.: A method of finite element tearing and interconnecting and its parallel solution algorithm. Internat. J. Numer. Methods Engrg. **32**, 1205–1227 (1991)
6. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**(2), 699–732 (2006)
7. Gander, M.J., Halpern, L.: Optimized Schwarz waveform relaxation for advection reaction diffusion problems. SIAM J. Numer. Anal. **45**, 666–697 (2007)
8. Gander, M.J., Halpern, L., Japhet, C., Martin, V.: Advection diffusion problems with pure advection approximation in subregions. In: Domain Decomposition Methods in Science and Engineering XVI, Lect. Notes Comput. Sci. Eng., 55, pp. 239–246. Springer, Berlin (2007)
9. Gander, M.J., Halpern, L., Nataf, F.: Optimized Schwarz waveform relaxation for the one dimensional wave equation. SIAM J. Numer. Anal. **41**(5), 1643–1681 (2003)
10. Gander, M.J., Kwok, F., Mandal, B.C.: Dirichlet–Neumann and Neumann–Neumann waveform relaxation methods for the time-dependent heat equation. In preparation
11. Gander, M.J., Stuart, A.: Space-time continuous analysis of waveform relaxation for the heat equation. SIAM J. Sci. Comput. **19**(6), 2014–2031 (1998)
12. Giladi, E., Keller, H.B.: Space-time domain decomposition for parabolic problems. Numer. Math. **93**, 279–313 (2002)
13. Lelarasmee, E., Ruehli, A., Sangiovanni-Vincentelli, A.: The waveform relaxation method for time-domain analysis of large scale integrated circuits. IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **1**(3), 131–145 (1982)
14. Lions, P.L.: On the Schwarz alternating method. I. In: First International Symposium on Domain Decomposition Methods for Partial Differential Equations (1989)
15. Martin, V.: An optimized Schwarz waveform relaxation method for the unsteady convection diffusion equation in two dimensionsht. Appl. Numer. Math. **52**(4), 401–428 (2005)
16. Miekkala, U., Nevanlinna, O.: Convergence of dynamic iteration methods for initial value problems. SIAM J. Sci. and Stat. Comput. **8**, 459482 (1987)
17. Oberhettinger, F., Badii, L.: Tables of Laplace Transforms. Springer-Verlag (1973)

GPU-based Parallel Reservoir Simulators

Zhangxin Chen[1], Hui Liu[1], Song Yu[1], Ben Hsieh[1] and Lei Shao[1]

1 Introduction

Nowadays reservoir simulators are indispensable tools to reservoir engineers. They are widely used in the optimization and prediction of oil and gas production. However, for large-scale reservoir simulation, computational time is usually too long. A case with over one million grid blocks may run weeks or even months. High performance processors and well-designed software are demanded. Though today's CPUs (Central Processing Unit) are much more powerful than before, performance of single CPU tends to slow down due to material and energy consumption and heat dissipation issues. Processor vendors have begun to move to multiple processing units, which form two major directions: multi-core CPUs and many-core GPUs [11].

In reservoir simulation, numerical methods like the finite difference and finite volume methods [7] are often used to discretize the mathematical models. Linear and nonlinear systems arising from the discretized models by those methods are sparse, which are usually time-consuming and difficult to solve. Krylov subspace solvers [18, 1] are general methods to solve these linear systems, and for large-scale reservoir simulation with over one million grid blocks, a reservoir simulator may take 90% or even more time on the solution of the linear systems. Fast and accurate linear and nonlinear solvers are essential to reservoir simulators. Saad et al. developed the GMRES solver for general unsymmetric linear systems [1, 18] and Vinsome designed the ORTHOMIN solver, which was originally developed for reservoir simulators [19]. PCG, BICGSTAB, algebraic multigrid and direct linear solvers were also proposed. Commonly used preconditioners were also developed, such as Incomplete LU (ILU) factorization, domain decomposition, algebraic multigrid, and multi-stage preconditioners [1, 18]. GPUs (Graphics Processing Unit) are usually used for display. Since each pixel can be processed simultaneously, GPUs are designed in such a way that they can manipulate data in parallel. Their float point performance and memory speed are very high [16, 15]. In general, GPUs are ten times faster than general CPUs [16, 15], which makes them powerful devices for parallel computing. Since GPUs are designed for graphics processing and not for general tasks, their architectures are different from those of CPUs. Hence new algorithms for GPUs should be developed to utilize GPUs' performance. NVIDIA developed a hybrid matrix format, the corresponding sparse matrix-vector multiplication kernel and a GPU-based linear solver package CUSP [4, 5, 2]. Bell et al. from

[1]Department of Chemical and Petroleum Engineering, University of Calgary, 2500 University Drive NW, Calgary, AB, Canada, T2N 1N4, e-mail: {zhachen}{hui.j.liu}{soyu}{bhsieh}{lshao}@ucalgary.ca

NVIDIA also investigated fine-grained parallelism of AMG solvers using a single GPU [3]. Saad et al. developed a sparse matrix-vector multiplication kernel for JAD matrix format and the GMRES solver [11]. Chen et al. designed a new matrix format, HEC, a new matrix-vector multiplication kernel, Krylov subspace solvers, algebraic multigrid solvers and several preconditioners [20, 13, 14, 12]. Haase et al. developed a parallel AMG solver for a GPU cluster [10]. In this paper, we will introduce our work on developing a GPU-based parallel iterative linear solver package and applying it to reservoir simulation.

The framework is as follows: In §2.1, GPU computing, our parallel linear solvers and GPU-based reservoir simulators are introduced. In §3, numerical experiments are presented.

2 Parallel Reservoir Simulator

In this section, we will propose GPU-based parallel linear solvers and preconditioners, and apply these solvers to reservoir simulation.

2.1 GPU Computing

The NVIDIA Fermi GPU, Tesla C2070, has 14 SMs (Streaming Multi-processors), and each SM has 32 SPs (Streaming Processors). That's 448 streaming processors in total while a normal CPU has only 2, 4, 6 or 8 cores. The GPU architectures are being developed rapidly. Each SM of the new Tesla Kepler GPUs has 192 SPs, much more than Fermi GPUs. The Tesla Kepler K20X has 2688 processors in total. At this moment we are using Fermi GPUs. The NVIDIA Fermi GPU, Tesla C2070, has a peak performance of 1030G FLOPS in single precision and a peak performance of 515G FLOPS in double precision, which are around 10 times faster than that of CPUs. The Tesla Kepler K20X GPU has a peak performance of 1310G FLOPS in double precision and a peak performance of 3950G FLOPS in single precision [17].

Each SM has its own L1 cache, shared memory and register. They share L2 cache, constant memory, texture memory and global memory. The global memory stores most of data, and is used to communicate with CPUs. The NVIDIA Tesla C2070 has 6 GB memory. Its memory speed is around 144 GB/s while the memory speed of CPUs is around 15 GB/s. The GPU memory is also about 10 times faster than the CPU memory. The Tesla Kepler K20X GPU has a memory speed of 250GB/s [17]. NVIDIA provides CUDA Toolkit [16, 15] to help users develop high performance programs.

2.2 Parallel Linear Solvers

GPUs have different architectures from general purpose CPUs. The NVIDIA GPUs access global memory in a coalesced way, which means that if the memory access is arranged well, threads in a grid block can fetch data in one or a few rounds. In this case, the memory access speed is the highest and codes are efficient. GPUs are emerging parallel devices. However, algorithms that work well on CPUs may not work effectively on GPUs [12]. We develop a new matrix format and the corresponding sparse matrix-vector multiplication kernel (SPMV) to accelerate iterative linear solvers. The new matrix format, HEC (Hybrid of ELL and CSR format), is shown in Fig. 1. A HEC matrix has two submatrices, ELL matrix and CSR matrix. The ELL submatrix stores the regular part of a given matrix and the CSR submatrix stores the irregular part of the given matrix. The ELL submatrix is stored in column-major manner. The main advantage of HEC is that it is friendly to ILU-related preconditioners. When we store a lower triangular matrix, it's clear that the last element in ELL or CSR part is a diagonal element and elements before the diagonal one are easy to recognize. If we use HYB, which is efficient for SPMV, the irregular part is stored in a COO matrix, we don't know which element is the diagonal one or elements before it. The second advantage is that SPMV algorithm for HEC is simple and implementation is straightforward. The psudo-codes for SPMV operation is listed in Alg. 1. Alg. 1 is for calculating $y = Ax$. Other related BLAS 2 operations are similar. We also develop BLAS 1 operations. A typical operation $y = \alpha x + \beta y$ is shown in Alg. 2.

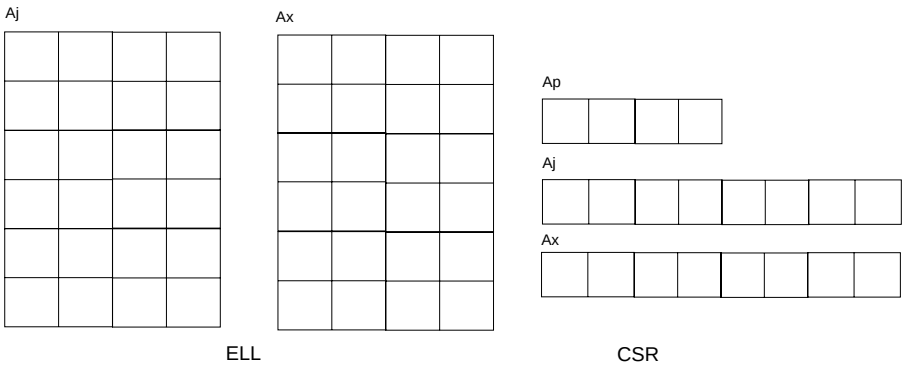


Fig. 1 HEC matrix format

We consider the following linear system:

$$Ax = b, \tag{1}$$

where A is a nonsingular $n \times n$ matrix, b is the right-hand side and x is the solution to be solved for. Krylov subspace solvers are general purpose methods for the solution

Algorithm 1 Sparse Matrix-Vector Multiplication, $y = Ax$

for i = 1: n **do**
  the *i*-th thread calculate the *i*-th row of ELL matrix; {ELL part, use one thread for each row}
end for

for i = 1: n **do**
  the *i*-th thread calculate the *i*-th row of CSR matrix; {CSR part, use one thread for each row}
end for

Algorithm 2 BLAS 1 subroutine, $y = \alpha x + \beta y$

for i = 1: n **do**
  $y[i] = \alpha x[i] + \beta y[i]$; {Use one GPU kernel to deal with this loop}
end for

of linear systems. Based on BLAS 1 and BLAS 2 subroutines we have implemented several Krylov subspace solvers and algebraic multigrid (AMG) solvers, including GMRES, CG, BICGSTAB, ORTHOMIN, classical AMG and smoothed aggregation AMG solvers.

In practice, an equivalent linear system of equation (1) is solved:

$$M^{-1}Ax = M^{-1}b, \tag{2}$$

where M is a nonsingular $n \times n$ matrix, called a preconditioner or left-preconditioner. When we choose a preconditioner M, a general principle is that M is an good approximation of A and in this case, it means that the product of M^{-1} and A approximates the unit matrix I well. The condition number of $M^{-1}A$ is much smaller than that of A and the preconditioned linear system (2) is much easier to solve compared to the original equation (1). Meanwhile, M should also be easy to construct and be easy to solve. We have implemented ILU(k)[18], block ILU(k) [18], ILUT(tol, p)[18], block ILUT(tol, p)[18], domain decomposition, approximate inverse, polynomial and algebraic multigrid preconditioners. For many preconditioners, an upper triangular linear system and a lower triangular linear system are required to solve. GPU-based parallel triangular solvers are developed to speed the solving of triangular linear systems. Details can be read in [13].

2.3 Reservoir Simulator

The reservoir simulator generates a Jacobian matrix in each Newton iteration. As mentioned above, the solution of linear systems in each Newton iteration may dominate the whole simulation time. For a large-scale black oil simulator, the linear solvers take over 90% of the running time. It is necessary for us to apply high performance linear solvers. We replace CPU-based linear solvers with GPU-based parallel linear solvers. The linearized systems are transferred to GPUs, and then GPUs

solve the linear system using hundreds of microprocessors in parallel and transfer the solution back to the simulator. By applying GPU-based linear solvers, reservoir simulators run much faster and it is possible for personal computers to run larger cases.

3 Numerical Experiments

Numerical experiments are performed on our workstation with Intel Xeon X5570 CPUs and NVIDIA Tesla C2050/C2070 GPUs. The operating system is CentOS 6.3 X86_64 with CUDA Toolkit 5.1 and GCC 4.4. All CPU codes are compiled with -O3 option and in this paper only one CPU core is employed. The type of float point number is double and blocks mean the number of sub-domains in this section.

Example 1. Several different SPMV algorithms [2, 4] are compared using matrices from the University of Florida sparse matrix collection [9]. Performance data [14] is collected in Tab 1 and numbers in the table mean speedup of GPU-based algorithms.

Table 1 Example 1: Performance of SPMV

Matrix	CSR	ELL	HYB	HEC
msc23052	0.71	1.55	2.29	2.50
cfd2	1.41	8.06	6.86	11.61
ESOC	1.06	11.56	11.56	11.61
cage13	1.25	7.56	9.42	11.04
af_shell8	1.17	8.66	9.63	11.12
parabolic_fem	4.36	9.97	7.56	10.00
Emilia_923	0.97	6.64	8.36	8.65
atmosmodd	2.94	14.54	14.50	14.57
Serena	0.98	1.79	7.26	7.29
SPE10	1.13	1.24	11.15	10.27

In Tab 1, the first column stands for matrix and others mean speedup using different SPMV algorithms. We can see that algorithms using HYB and HEC matrix formats are always efficient and the performance of our HEC matrix format is better than that of HYB.

Example 2. The matrix used here is from SPE10 [7, 8]. The SPE10 problem is a standard benchmark for the black oil simulator. The problem is highly heterogenous and it is designed to solve hard. The grid size for SPE10 is 60x220x85. The number of unknowns is 2,188,851 and the number of non-zeros is 29,915,573. The linear solver employed is GMRES(20), and block ILU(k), block ILUT(tol, p) and domain decomposition preconditioners are applied as preconditioners. Here RAS (Restricted Additive Schwarz) a domain decomposition method developed by Cai[6].

Performance data is collected in Table 2. For this example, loading time is always less than 1s.

Table 2 Example 2: Performance of SPE10

Preconditioner	Parameters	Setup (s)	CPU (s)	GPU (s)	Speedup
BILU(0)	(1, 0)	4.23	76.65	12.42	6.16
BILU(0)	(16, 0)	2.04	92.80	12.78	7.25
BILU(0)	(128, 0)	1.76	86.22	12.05	7.14
BILU(0)	(512, 0)	1.63	92.82	12.87	7.20
BILUT	(1, 0)	4.76	23.50	9.03	2.60
BILUT	(16, 0)	2.49	32.00	7.17	4.46
BILUT	(128, 0)	1.94	42.51	7.82	5.42
BILUT	(512, 0)	1.81	47.44	8.80	5.37
RAS	(256, 1)	9.28	106.61	14.36	7.41
RAS	(1024, 1)	11.91	110.36	16.36	6.73
RAS	(256, 2)	10.99	107.89	17.28	6.23
RAS	(1024, 2)	15.28	138.60	20.93	6.61

In Table 2, the first column stands for preconditioners. The second column stands for parameters used for each preconditioner and they are the number of sub-domains and overlap respectively. The others are for setup time, running time and speedup, respectively. From the table, we can speed ILU(0) 6.2 times faster. The speedup can be higher if we increase the number of blocks. The average speedup of BILU(0) is about 7. In this example, BILUT is the most effective preconditioner. It always takes the least running time. However, due to its irregular non-zero pattern, its speedup is lower than that of the other two precondtioners. Since there is not enough memory on GPU, the maximum number of blocks for RAS in this case is 1024. The average speedup of RAS is 6.5.

Example 3. The matrix used here is also from SPE10, which has the same size with the one we use in Example 2. Its pressure part has a dimension of 1,094,421 and has 7,478,141 non-zeros. Here we solve pressure matrix using algebraic multigrid solver and the entire matrix is solved by GMRES (40) with CPR-AMG (Constrained Pressure Residual) preconditioner [7]. To compare, the entire matrix is also solved using GMRES(40) with ILU(0) preconditioner. Classical AMG solver is applied here. The standard interpolator and damped Jacobi smoother are applied. V-cycle is used for solving phase and the coarsest level is solved using GMRES. The AMG solver has 8 levels. The termination criterium is 1e-6. Performance data is shown in Tab 3.

The speedup of AMG is 6.49 when solving pressure matrix. When CPR-AMG is used as a preconditioner, the GMRES(40) converges quickly. CPR-AMG is much more efficient than ILU(0). And a speedup of 6.74 is obtained. From Tab 3, we can also see that the setup phase takes too much time, which should be optimized in future.

Table 3 Example 3: Performance of AMG

Solver	Preconditioner	Setup (s)	CPU (s)	GPU (s)	Speedup	Residual	Iterations
AMG		3.96	6.86	1.04	6.49	4.88e-7	11
GMRES(40)	CPR-AMG	8.43	185.42	27.51	6.74	3.91e-7	9
GMRES(40)	ILU(0)	4.3	1037.26	195.96	5.29	9.42e-4	100

Example 4. This example is to test the speedup of our GPU solver in the whole black oil simulator. The case is SPE10 simulation in 100 days. The grid size for SPE10 is 60x220x85. The solver is GMRES(20). Performance data is shown in Tab 4.

Table 4 Example 4: Performance of black oil simulator

Preconditioner	Blocks	CPU (s)	GPU (s)	Speedup
BILU(0)	1	49610.28	7721.09	6.43
BILU(0)	4	53350.63	8524.31	6.26
BILU(0)	8	54286.07	8720.25	6.23
BILUT	1	19533.45	9008.22	2.17
BILUT	4	23187.85	8670.53	2.67
BILUT	8	21718.45	7908.42	2.75

As shown in Tab 4, the block ILU(0) preconditioner achieves a speedup of 6.2 while the speedup of block ILUT is much lower. The reason is that the non-zero pattern of ILUT is irregular and the ILU factorization takes too much time.

4 Conclusion

We have developed a GPU-based parallel linear solver package. When solving matrices from reservoir simulation, the parallel solvers are much more efficient than CPU-based linear solvers. However, efforts should be made to improve the setup phase of domain decomposition, the factorization of ILUT and parallelism of block ILUT preconditioner.

Acknowledgements The support of Department of Chemical and Petroleum Engineering, University of Calgary and Reservoir Simulation Group is gratefully acknowledged. The research is partly supported by NSERC/AIEES/Foundation CMG and AITF Chairs.

References

1. Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., C., R., der Vorst H., V.: Templates for the solution of linear systems: building blocks for iterative methods, 2nd Edition. SIAM (1994)
2. Bell N.and Garland, M.: Cusp: Generic parallel algorithms for sparse matrix and graph computations. (2012). URL `http://cusp-library.googlecode.com`. Version 0.3.0
3. Bell, N., Dalton, S., Olson, L.: Exposing fine-grained parallelism in algebraic multigrid methods. (2011)
4. Bell, N., Garland, M.: Efficient sparse matrix-vector multiplication on CUDA. (2008)
5. Bell, N., Garland, M.: Implementing sparse matrix-vector multiplication on throughput-oriented processors. In: Proc. Supercomputing (2009)
6. Cai X.-C.and Sarkis, M.: A restricted additive schwarz preconditioner for general sparse linear systems. SIAM J. Sci. Comput. pp. 792–797
7. Chen, Z., Huan, G., Ma, Y.: Computational Methods for Multiphase Flows in Porous Media. SIAM (2006)
8. Christie, M., Blunt, M.: Tenth spe comparative solution project: A comparison of upscaling techniques. SPE Reservoir Engineering and Evaluation. pp. 308–317
9. Davis, T.A.: University of florida sparse matrix collection, na digest. (1994)
10. Haase, G., Liebmann, M., Douglas, C.C., Plank, G.: A parallel algebraic multigrid solver on graphics processing units. HIGH PERFORMANCE COMPUTING AND APPLICATIONS. pp. 38–47 (2010)
11. Klie, H., Sudan, H., Li, R., Saad, Y.: Exploiting capabilities of many core platforms in reservoir simulation. In: SPE RSS Reservoir Simulation Symposium (2011)
12. Liu, H., Yu, S., Chen, Z.: Development of algebraic multigrid solvers using gpus. In: SPE RSS Reservoir Simulation Symposium. (2012)
13. Liu, H., Yu, S., Chen, Z., Hsieh, B., Shao, L.: Parallel preconditioners for reservoir simulation on gpu. In: SPE Latin America and Caribbean Petroleum Engineering Conference. (2012)
14. Liu, H., Yu, S., Chen, Z., Hsieh, B., Shao, L.: Sparse matrix-vector multiplication on nvidia gpu. International Journal of Numerical Analysis and Modeling. (2012)
15. NVIDIA: Cuda c best practices guide (version 3.2). (2010)
16. NVIDIA: Nvidia cuda programming guide (version 3.2). (2010)
17. NVIDIA: Nvidia tesla gpu products. (2012). URL `http://www.nvidia.com/object/tesla-servers.html`
18. Saad, Y.: Iterative methods for sparse linear systems (2nd edition). SIAM (2003)
19. Vinsome, P.: an iterative method for solving sparse sets of simultaneous linear equations. In: SPE Symposium on Numerical Simulation of Reservoir Performance (1976)
20. Yu, S., Liu, H., Chen, Z., Hsieh, B., Shao, L.: Gpu-based parallel reservoir simulation for large-scale simulation problems. In: SPE EAGE Annual Conference & Exhibition, SPE-152271 (2012)

Optimized Schwarz methods with overlap for the Helmholtz equation

Martin J. Gander[1] and Hui Zhang[1]

1 Introduction

For the Helmholtz equation, simple absorbing conditions of the form $\partial_n - i\omega$ were proposed as transmission condition (TC) in Schwarz methods first without overlap in [4], and later also with overlap, see [3, 12]. More advanced TCs can also be used, see e.g. [11, 14, 2]. Furthermore, parameters can be introduced into TCs and then optimized for rapid convergence, which led to the so called optimized Schwarz methods, see e.g. [6, 13] for elliptic equations. *Without* overlap, the parameters involved in some zero- and second-order TCs for the Helmholtz equation have been optimized in [8, 7]. *With* overlap, preliminary numerical studies of the parameters have been presented in [5, 9]. In this paper, we present the asymptotic solutions of the corresponding optimization problems with small overlap. We also compare the optimized parameters with other choices based on convergence factors and actual iteration numbers. We finally test for the first time Taylor second-order absorbing conditions for domain decomposition *with* overlap in the Helmholtz case.

2 Schwarz Methods with Overlap

As a model problem, we consider the Helmholtz equation in free space,

$$(\omega^2 + \Delta)u = f(x,y), \quad (x,y) \in \mathbb{R} \times \mathbb{R}^{d-1},$$

equipped with the Sommerfeld radiation condition

$$\lim_{r\to\infty} r^{\frac{d-1}{2}}(\frac{\partial u}{\partial r} - i\omega) = 0, \quad r = \sqrt{x^2 + \sum_{j=1}^{d-1} y_j^2}.$$

We decompose the domain into two overlapping subdomains $\Omega_1 = (-\infty, L) \times \mathbb{R}^{d-1}$ and $\Omega_2 = (0,\infty) \times \mathbb{R}^{d-1}$ with the overlap size $L > 0$. The Schwarz iteration reads

$$\begin{aligned} \omega^2 u_1^{n+1} + \Delta u_1^{n+1} &= f(x,y), && (x,y) \in \Omega_1, \\ (\partial_x + \mathscr{S}_1)(u_1^{n+1})(L,y) &= (\partial_x + \mathscr{S}_1)(u_2^n)(L,y), && y \in \mathbb{R}^{d-1}, \end{aligned}$$

[1]Section of Mathematics, University of Geneva, 2-4 rue du Liévre, Case postale 64, e-mail: {martin.gander}{hui.zhang}@unige.ch

and

$$\begin{aligned} \omega^2 u_2^{n+1} + \Delta u_2^{n+1} &= f(x,y), && (x,y) \in \Omega_2, \\ (-\partial_x + \mathscr{S}_2)(u_2^{n+1})(0,y) &= (-\partial_x + \mathscr{S}_2)(u_1^n)(0,y), && y \in \mathbb{R}^{d-1}, \end{aligned}$$

where $\mathscr{S}_j$, $j = 1,2$ are two linear operators in some trace spaces along $\{L\} \times \mathbb{R}^{d-1}$ and $\{0\} \times \mathbb{R}^{d-1}$, respectively. For the analysis it suffices to consider by linearity the case $f(x,y) = 0$ and to analyze convergence to the zero solution. We take a Fourier transform in the y direction to obtain

$$\begin{aligned} (\omega^2 - |k|^2)\hat{u}_1^{n+1} + \partial_{xx}^2 \hat{u}_1^{n+1} &= 0, && x \in (-\infty, L), \\ (\partial_x + s_1)(\hat{u}_1^{n+1})(L,k) &= (\partial_x + s_1)(\hat{u}_2^n)(L,k), \end{aligned}$$

and

$$\begin{aligned} (\omega^2 - |k|^2)\hat{u}_2^{n+1} + \partial_{xx}^2 \hat{u}_2^{n+1} &= 0, && x \in (0, \infty), \\ (-\partial_x + s_2)(\hat{u}_2^{n+1})(0,k) &= (-\partial_x + s_2)(\hat{u}_1^n)(0,k), \end{aligned}$$

where k is the Fourier variable of y and s_j denotes the symbol of $\mathscr{S}_j$. Since the Sommerfeld radiation condition excludes growing solutions as well as incoming modes at infinity we obtain the solutions

$$\begin{aligned} \hat{u}_1^{n+1}(x,k) &= \hat{u}_1^{n+1}(L,k) e^{\lambda(k)(x-L)}, \\ \hat{u}_2^{n+1}(x,k) &= \hat{u}_2^{n+1}(0,k) e^{-\lambda(k)x}, \end{aligned}$$

where $\lambda(k)$ denotes the root of the characteristic equation $\lambda^2 + (\omega^2 - |k|^2) = 0$ with positive real part or negative imaginary part,

$$\lambda(k) := \begin{cases} \sqrt{|k|^2 - \omega^2} & \text{for } |k| > \omega, \\ -i\sqrt{\omega^2 - |k|^2} & \text{for } |k| < \omega. \end{cases} \tag{1}$$

Substitution of the solutions into the transmission conditions yields

$$\begin{aligned} \hat{u}_1^{n+1}(L,k) &= \tfrac{s_1(k) - \lambda(k)}{s_1(k) + \lambda(k)} e^{-\lambda(k)L} \hat{u}_2^n(0,k), \\ \hat{u}_2^{n+1}(0,k) &= \tfrac{s_2(k) - \lambda(k)}{s_2(k) + \lambda(k)} e^{-\lambda(k)L} \hat{u}_1^n(L,k). \end{aligned}$$

By recursion we have $\hat{u}_1^{n+1}(L,k) = \rho(k)\hat{u}_1^{n-1}(L,k)$ and $\hat{u}_2^{n+1}(0,k) = \rho(k)\hat{u}_2^{n-1}(0,k)$, where the convergence factor ρ for a double iteration is defined by

$$\rho(k) = \frac{s_1(k) - \lambda(k)}{s_1(k) + \lambda(k)} \cdot \frac{s_2(k) - \lambda(k)}{s_2(k) + \lambda(k)} e^{-2\lambda(k)L}. \tag{2}$$

Setting the two complex parameters $s_1 = p_1 - iq_1$ and $s_2 = p_2 - iq_2$, with $p_j, q_j \in \mathbb{R}$, and inserting s_1, s_2 and (1) into the convergence factor (2), we find after simplifying

$$|\rho(p_1,q_1,p_2,q_2,k)|^2 = \begin{cases} \frac{p_1^2+(q_1-\sqrt{\omega^2-|k|^2})^2}{p_1^2+(q_1+\sqrt{\omega^2-|k|^2})^2}\,\frac{p_2^2+(q_2-\sqrt{\omega^2-|k|^2})^2}{p_2^2+(q_2+\sqrt{\omega^2-|k|^2})^2}, & |k|^2<\omega^2, \\ \frac{q_1^2+(p_1-\sqrt{|k|^2-\omega^2})^2}{q_1^2+(p_1+\sqrt{|k|^2-\omega^2})^2}\,\frac{q_2^2+(p_2-\sqrt{|k|^2-\omega^2})^2}{q_2^2+(p_2+\sqrt{|k|^2-\omega^2})^2}e^{-4\lambda(k)L}, & |k|^2>\omega^2. \end{cases} \tag{3}$$

As long as $|k| \neq \omega$ and $p_j, q_j > 0$, we have $|\rho| < 1$.

Remark 1. It was shown in [6] that the two-sided operators $\mathscr{S}_j = s_j \in \mathbb{C}$ can be transformed into the second-order operators

$$\tilde{\mathscr{S}}_1 = \tilde{\mathscr{S}}_2 = r_1 - r_2\nabla_y^2, \quad \text{with} \quad r_1 = \frac{-\omega^2 + s_1 s_2}{s_1+s_2}, \quad r_2 = \frac{1}{s_1+s_2}, \tag{4}$$

and the associated convergence factor for a single iteration is then given by

$$\tilde{\rho}(k) = \frac{s_1(k)-\lambda(k)}{s_1(k)+\lambda(k)} \cdot \frac{s_2(k)-\lambda(k)}{s_2(k)+\lambda(k)} e^{-\lambda(k)L}, \tag{5}$$

which is just (2) with L replaced by $L/2$.

3 Optimized transmission conditions

For simplicity, we consider $p_1 = q_1$, $p_2 = q_2$. Our goal is to find good parameters p_1, p_2 such that the modulus of the squared convergence factor (3) is as small as possible over a range of frequencies $|k| \in [k_{\min}, k_-] \cup [k_+, k_{\max}]$, where $k_- < \omega < k_+$. We require $|k|$ to be away from ω because $|\rho| = 1$ when $|k| = \omega$, independently of what one chooses for the parameters p_j and q_j. Since in general we do not know how the Fourier coefficients of the initial error are distributed over the frequencies, we minimize $|\rho|$ for the worst case, that is, we solve the min–max problem

$$\operatorname{argmin}_{(p_1,p_2)\in\mathbb{P}} \left(\max_{|k|\in[k_{\min},k_-]\cup[k_+,k_{\max}]} |\rho(p_1,p_1,p_2,p_2,k)|^2 \right), \tag{6}$$

where $\mathbb{P}$ is a certain search domain of the parameters. For well-posedness of the subdomain problems, we should choose $\mathbb{P} \subset [0,\infty)^2$. The best approximation problem (6) is difficult to solve, and we only give asymptotic formulas for the parameters such that the convergence factor is as small as possible in different limiting processes in the mesh size h and the wave number ω.

The proofs of the following theorems are beyond the scope of this short paper and will appear in [10].

Theorem 1. *Let $L = C_L h$, $k_{\max} \in [C/h, \infty]$, $C_L, C, k_{\min}, k_-, k_+$ and ω be positive and independent of h, $k_{\min} < k_- < \omega$, $k_{\max} > k_+ > \omega$ and $\mathbb{P} = \{(p_1,p_2) \,|\, 0 \le p_1 \le p_2 < \infty\}$. Suppose h is small and $|k| \in [k_{\min}, k_-] \cup [k_+, k_{\max}]$. If we set*

$$\begin{aligned} p_1 &= p_1^* = C_\omega^{2/5}(4L)^{-1/5}/2 + o(h^{-1/5}), \\ p_2 &= p_2^* = C_\omega^{1/5}(4L)^{-3/5} + o(h^{-3/5}), \end{aligned} \tag{7}$$

where $C_\omega = \min\left\{\omega^2 - k_-^2, k_+^2 - \omega^2\right\}$*, then (3) is bounded by* $1 - 4(4L\sqrt{C_\omega})^{1/5} + o(h^{1/5})$*. Moreover, any solution of (6) must satisfy (7).*

Theorem 2. *Let* $L = C_L h$*,* $h = C_h/\omega^\gamma$*,* $\gamma \geq 1$*,* $k_{\max} \in [C/h, \infty]$*,* $\delta_\omega = \min\{\omega - k_-, k_+ - \omega\}$ *with* C_L, C_h, C, k_- *and* k_+ *positive constants independent of* ω*,* $k_{\min} < k_- < \omega$*,* $k_{\max} > k_+ > \omega$ *and* $\mathbb{P} = \{(p_1, p_2) \,|\, 0 \leq p_1 \leq p_2 < \infty\}$*. Suppose* ω *is large and* $|k| \in [k_{\min}, k_-] \cup [k_+, k_{\max}]$*. Then, for* $1 \leq \gamma < 9/8$ *any solution of (6) satisfies*

$$\begin{aligned} p_1^* &= \delta_\omega^{3/8}(\omega/2)^{5/8} + o(\omega^{5/8}), \\ p_2^* &= (2\delta_\omega)^{1/8}\omega^{7/8} + o(\omega^{7/8}), \end{aligned}$$

for which (3) is bounded by $1 - 4 \cdot 2^{1/8}\delta_\omega^{1/8}\omega^{-1/8} + o(\omega^{-1/8})$*. For* $\gamma > 9/8$*, any solution of (6) satisfies*

$$\begin{aligned} p_1^* &= (\delta_\omega \omega)^{2/5} L^{-1/5}/2 + o(\omega^{2/5+\gamma/5}), \\ p_2^* &= (\delta_\omega \omega)^{1/5} L^{-3/5}/2 + o(\omega^{1/5+3\gamma/5}), \end{aligned}$$

and (3) is bounded by $1 - 4\sqrt{2}(C_h C_L)^{1/5}\delta_\omega^{1/10}\omega^{1/10-\gamma/5} + o(\omega^{1/10-\gamma/5})$*. Finally, for* $\gamma = 9/8$*, any solution of (6) satisfies*

$$\begin{aligned} p_1^* &= (C_h C_L \delta_\omega)^{1/3}\omega^{5/8} + o(\omega^{5/8}), \\ p_2^* &= C_h C_L\,\omega^{7/8} + o(\omega^{7/8}), \end{aligned}$$

and (3) is bounded by

$$\begin{cases} 1 - 16 C_h C_L\,\omega^{-1/8} + o(\omega^{-1/8}), & \text{if } 2^{-15/8}\delta_\omega^{1/8} \leq C_h C_L, \\ 1 - 2\sqrt{2}\,\delta_\omega^{1/6} C_h^{-1/3} C_L^{-1/3}\,\omega^{-1/8} + o(\omega^{-1/8}), & \text{if } 2^{-15/8}\delta_\omega^{1/8} \geq C_h C_L. \end{cases}$$

Remark 2. In the particular case $\gamma = 9/8$, the constant in front of the leading term of p_1^* can be an arbitrary number in the interval $[\sqrt{2\,\delta_\omega}/(8C_hC_L), 32C_h^3C_L^3]$ in order to solve (6). But the choice in the above theorem is the best in the sense that it simultaneously minimizes the maximum of the other local but not global maxima.

Remark 3. In practice, we use only the leading order terms of the optimized parameters. But it is also possible to extract higher order terms.

Fig.1 shows the convergence factors of different Schwarz methods, obtained for the model problem in $\mathbb{R}^2$, with $\omega = 20\pi$ and $h = 1/100$. The maximum of the convergence factors for double iterations over the chosen interval $k \in [\pi, \omega - \pi] \cup [\omega + \pi, \pi/h]$ are 1.0, for the classical Schwarz method and Després' method without overlap [4], 0.4376 for Després' method with overlap [3, 12], 0.1548 for the optimized Schwarz methods without overlap [7], and and 0.0764 for the same method with

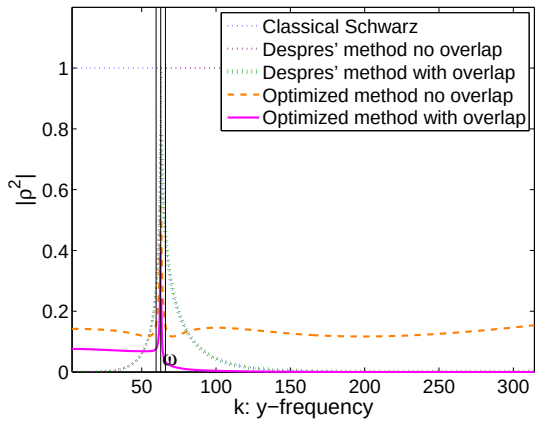


Fig. 1 Convergence factors of different Schwarz methods as functions of the Fourier parameter k, for $\omega = 20\pi$, $h = 1/100$. The vertical lines indicate the ω, $\omega - \pi$ and $\omega + \pi$ which are used to exclude a short interval for the optimized methods.

overlap. The overlap size we chose is $2h$, and we used the second-order formulation (4), (5) for the optimized methods.

4 Numerical experiments

We used the ORAS formulation described in [13] for our implementation. As an alternative, one could also use a substructured formulation, see e.g. [9]. We implemented the second-order transmission conditions as indicated in Remark 1. We always solve the homogeneous equation with the zero solution and use a *random* initial guess to stimulate all frequencies. We use the domain decomposition $\Omega_1 = (0, \frac{1}{2} + h) \times (0,1)$, $\Omega_2 = (\frac{1}{2} - h, 1) \times (0,1)$, and iterate until the relative residual is less than 10^{-8}. We compare the overlapping Schwarz methods with optimized second-order transmission condition denoted by OO2 to those with the classical Dirichlet condition denoted by Cl, simple absorbing conditions of the form $\partial_n - i\omega$ (i.e. Després' method with overlap, c.f. [3, 12]) denoted by TO0, because it corresponds to a Taylor expansion of zero order of the symbol of the DtN operator, and the second-order low frequency absorbing condition, which is denoted by TO2. Since the Schwarz methods can be used as a stationary iterative solver, or as a preconditioner for GMRES, both cases are tested, except for the classical Schwarz stationary iteration, which can not converge.

We consider the open cavity problem with homogeneous Dirichlet boundary conditions on the top and the bottom of the unit square and the TO2 second-order absorbing conditions [1] on the left and the right sides, and also the free space problem

Table 1 Iteration numbers for the open cavity problem on the left and for the free space problem on the right, top half for $\omega = 9.5\pi$, and below for $\omega = 10\pi$.

$1/h$	Stationary			GMRES				Stationary			GMRES			
	TO0	TO2	OO2	Cl.	TO0	TO2	OO2	TO0	TO2	OO2	Cl.	TO0	TO2	OO2
50	34	35	14	25	16	15	12	23	24	17	25	17	15	13
100	74	84	17	30	22	22	13	33	41	21	28	21	21	14
200	166	172	20	38	27	32	14	51	73	22	33	27	30	15
400	343	345	20	49	33	41	14	85	135	23	42	33	40	15
800	662	717	21	67	40	50	16	144	249	24	58	42	49	16
50	67	70	19	26	15	14	14	22	23	17	26	16	15	13
100	227	222	31	30	21	22	15	32	40	20	27	21	21	14
200	469	371	44	38	28	32	15	50	71	22	33	27	30	15
400	681	455	51	51	34	42	15	83	130	22	43	34	40	15
800	864	504	55	68	41	52	17	136	241	23	55	42	49	15

truncated to the unit square with the TO2 second-order absorbing conditions at the boundary.

First, we fix $\omega = 9.5\pi$ (or $\omega = 10\pi$) which are away from (or on) the sine frequencies at the continuous level in the y-direction. The corresponding iteration numbers are listed in Table 1. We can see that the minimum distance from ω to the frequencies at the discrete level in the y-direction plays an important role in all the *stationary* iterations while in the *GMRES* iterations this effect is only moderate. Fig.2 shows the asymptotic behavior of the different Schwarz methods as $h \to 0$, for the open cavity problem, and confirms our Fourier analysis results in Theorem 1.

Now we fix $h\omega$ or $h\omega^{3/2}$ constant to see how the Schwarz methods behave for higher and higher wave numbers, which corresponds to Theorem 2. The iteration numbers are listed in Table 2. We see that the optimized method still converges faster

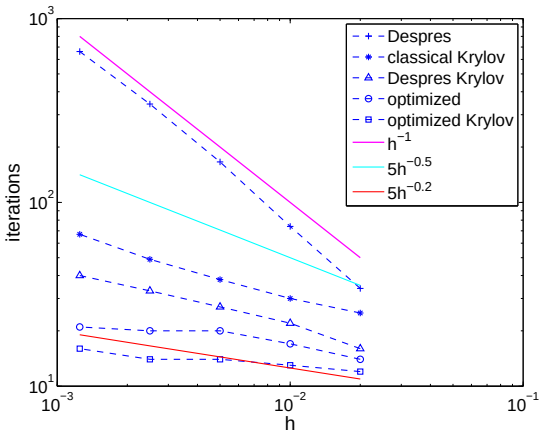


Fig. 2 Asymptotic behavior of the Schwarz methods for the open cavity, $\omega = 9.5\pi$.

Table 2 Iteration numbers for the open cavity problem on the left and for the free space problem on the right, top half for $h\omega = \pi/5$, and below for $h\omega^{3/2} \approx 3.52$.

$1/h$	Stationary			GMRES				Stationary			GMRES			
	TO0	TO2	OO2	Cl.	TO0	TO2	OO2	TO0	TO2	OO2	Cl.	TO0	TO2	OO2
100	86	67	35	38	20	19	16	27	27	18	35	20	18	15
200	–	110	–	48	25	22	19	33	30	19	43	25	21	16
400	280	150	72	69	38	32	20	43	37	19	53	28	25	17
800	178	139	44	76	42	35	25	56	45	19	65	32	30	17
100	80	87	15	34	22	20	14	29	31	19	30	21	18	14
200	–	2948	–	43	27	27	19	42	39	19	34	27	24	15
400	266	279	26	49	33	32	18	56	50	20	41	35	30	16
800	208	218	21	70	46	41	18	78	65	20	47	43	37	16

than the others when used as a preconditioner for GMRES, while the stationary iterations are again greatly affected by the discrete frequencies close to the wave number. The bars in the tables represent divergence.

Next, we test the various Schwarz methods for an increasing number of subdomains. Since in most cases the stationary iterations diverge, we only show the GMRES iteration numbers in Table 3, where we use a bar to represent iteration numbers larger than 3000. We can see, neglecting the numbers in the parentheses for the moment, that all the methods deteriorate rapidly and the overlapping TO2 method outperforms the others eventually. Clearly the optimization of the two subdomain convergence factor does not predict well the optimal choice in the case of many subdomains for the Helmholtz equation.

To partially improve the OO2 method, we introduce now two heuristics. First, we take $\delta_\omega = N\pi/2$ instead of $\delta_\omega = \pi$ in the former experiments, where N denotes the number of subdomains in the x-direction. Second, since the real parts of the parameters slow down the convergence for propagating modes, which becomes worse when the number of subdomains increases, we use $s_j = (2/N - i)\, p_j$ $(j = 1,2)$ instead of $s_j = (1-i)\, p_j$. The new results are shown in the parentheses of Table 3, where the first numbers are obtained by using the two heuristics and the second numbers are from numerically optimized parameters based on a new many-subdomain Fourier

Table 3 Iteration numbers of GMRES, $h = 1/256$, $\omega = 51.2\pi$, overlap $2h$.

Sub.	open cavity						free space					
	Cl.	TO0	TO2	OO2			Cl.	TO0	TO2	OO2		
2×1	52	28	24	18			48	25	23	16		
4×1	396	68	46	68	(45	40)	163	29	24	45	(30	22)
8×1	–	160	102	162	(91	88)	–	44	33	108	(50	36)
16×1	–	682	221	492	(183	188)	–	88	65	258	(82	67)
2×2	118	66	63	61			49	27	25	20		
4×4	2192	184	172	183	(177	166)	372	38	33	49	(42	35)
8×8	–	789	618	734	(638	601)	–	69	65	104	(82	70)
16×16	–	2047	1473	2268	(1859	1514)	–	123	127	184	(168	136)

analysis. But still, the low frequency Taylor conditions perform best in these experiments. Our on-going work is to take a closer look at the multi-domain case and to seek better choices of parameters if it is possible.

Acknowledgements The work was supported by the University of Geneva. The second author was also partially supported by the International Science and Technology Cooperation Program of China (2010DFA14700).

References

1. Bamberger, A., Joly, P., Roberts, J.E.: Second-order absorbing boundary conditions for the wave equation: a solution for the corner problem. SIAM J. Numer. Anal. **27**, 323–352 (1990)
2. Boubendir, Y., Antoine, X., Geuzaine, C.: A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation. J. Comput. Phys. **231**, 262–280 (2012)
3. Cai, X.C., Casarin, M.A., Elliott Jr., F.W., Widlund, O.B.: Overlapping Schwarz algorithms for solving Helmholtz's equation. In: J. Mandel, C. Farhat, X.C. Cai (eds.) Domain Decomposition Methods 10, Contemporary Mathematics 218, Boulder, pp. 437–445. AMS (1998)
4. Després, B.: Domain decomposition method and the Helmholtz problem. In: G.C. Cohen, L. Halpern, P. Joly (eds.) Mathematical and numerical aspects of wave propagation phenomena, Strasbourg, pp. 44–52. SIAM (1991)
5. Gander, M.J.: Optimized Schwarz methods for Helmholtz problems. In: N. Debit, M. Garbey, R.H.W. Hoppe, et. al. (eds.) Domain Decomposition Methods in Science and Engineering, 13th International Conference on Domain Decomposition Methods, Barcelona, pp. 247–254. CIMNE (2002)
6. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**, 699–731 (2006)
7. Gander, M.J., Halpern, L., Magoulès, F.: An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation. Int. J. Numer. Meth. Fluids **55**, 163–175 (2007)
8. Gander, M.J., Magoulès, F., Nataf, F.: Optimized Schwarz methods without overlap for the Helmholtz equation. SIAM J. Sci. Comput. **24**, 38–60 (2002)
9. Gander, M.J., Zhang, H.: Domain decomposition methods for the Helmholtz equation: a numerical investigation. In: R. Bank, M. Holst, J. Xu (eds.) Domain Decomposition Methods in Science and Engineering XX, Lecture Notes in Computational Science and Engineering, San Diego, p. 215–222. Springer-Verlag (2012)
10. Gander, M.J., Zhang, H.: Optimized Schwarz methods with overlap for the Helmholtz equation. *in preparation* (2014)
11. Ghanemi, S.: A domain decomposition method for Helmholtz scattering problems. In: P.E. Bjorstad, M.S. Espedal, D.E. Keyes (eds.) Proceedings of the 9th Intl. Conf. on Domain Decomposition Methods, Ullensvang, pp. 105–112. ddm.org (1998)
12. Kimn, J.H., Sarkis, M.: Restricted overlapping balancing domain decomposition methods and restricted coarse problems for the Helmholtz problem. Comput. Methods Appl. Mech. Engrg. **196**, 1507–1514 (2007)
13. St-Cyr, A., Gander, M.J., Thomas, S.J.: Optimized multiplicative, additive, and restricted additive Schwarz preconditioning. SIAM J. Sci. Comput. **29**, 2402–2425 (2007)
14. Stupfel, B.: Improved transmission conditions for a one-dimensional domain decomposition method applied to the solution of the Helmholtz equation. J. Comput. Phys. **229**, 851–874 (2010)

DG discretization of optimized Schwarz methods for Maxwell's equations

Mohamed El Bouajaji[1], Victorita Dolean[2], Martin J. Gander[3], Stéphane Lanteri[1], and Ronan Perrussel[4]

1 Introduction

In the last decades, Discontinuous Galerkin (DG) methods have seen rapid growth and are widely used in various application domains (see [13] for an historical introduction). This is due to their main advantage of combining the best of finite element and finite volume methods. For the time-harmonic Maxwell equations, once the problem is discretized with a DG method, finding robust solvers is a difficult task since one has to deal with indefinite problems. From the pioneering work of Després [5] where the first provably convergent domain decomposition (DD) algorithm for the Helmholtz equation was proposed and then extended to Maxwell's equations in [6], other studies followed. Preliminary attempts to obtain better algorithms for this kind of equations were given in [3, 4, 12], where the first ideas of optimized Schwarz methods can be found. Then, the advantage of the optimization process was used for the second order Maxwell system in [1]. Later on, an entire hierarchy of optimized transmission conditions for the first order Maxwell's equations was proposed in [9, 11] . For the second order or curl-curl Maxwell's equations second order optimized transmission conditions can be found in [14, 15, 16, 17]. We study here optimized Schwarz DD methods for the time-harmonic Maxwell equations discretized by a DG method. Due to the particularity of the latter, DG discretization applied to more sophisticated Schwarz methods is not straightforward. In this work we show a strategy of discretization and prove the equivalence between multi-domain and single-domain solutions. The proposed discrete framework is then illustrated by some numerical results in the two-dimensional case.

We consider time-harmonic Maxwell's equations in a homogeneous medium written as a first order system (see [10] for more details)

$$G_0\mathbf{W} + G_x\partial_x\mathbf{W} + G_y\partial_y\mathbf{W} + G_z\partial_z\mathbf{W} = 0, \tag{1}$$

where

$$\mathbf{W} = \begin{pmatrix} \mathbf{E} \\ \mathbf{H} \end{pmatrix}, G_0 = \begin{pmatrix} (\sigma + i\omega)\mathbb{I}_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & i\omega\mathbb{I}_{3\times 3} \end{pmatrix}$$

with $\mathbf{E}$, $\mathbf{H}$ the complex-valued electric and magnetic fields, ω the angular frequency of the time-harmonic wave, σ the electric conductivity. For a general vector $\mathbf{n} =$

[1] Inria Sophia Antipolis-Méditerranée, France e-mail: {Mohamed.El_Bouajaji}{Stephane.Lanteri}@inria.fr .[2] University of Nice-Sophia Antipolis, France e-mail: dolean@unice.fr .[3] University of Geneva, Switzerland e-mail: martin.gander@unige.ch .[4] CNRS, Université de Toulouse, Laplace, France e-mail: perrussel@laplace.univ-tlse.fr

$\begin{pmatrix} n_x & n_y & n_z \end{pmatrix}$, we also define the matrices

$$G_{\mathbf{n}} = \begin{pmatrix} 0_{3\times 3} & N_{\mathbf{n}} \\ N_{\mathbf{n}}^T & 0_{3\times 3} \end{pmatrix} \text{ and } N_{\mathbf{n}} = \begin{pmatrix} 0 & n_z & -n_y \\ -n_z & 0 & n_x \\ n_y & -n_x & 0 \end{pmatrix}.$$

Then, for $l \in \{x,y,z\}$, we have that $N_l = N_{\mathbf{e}_l}$ and $G_l = G_{\mathbf{e}_l}$, where $\mathbf{e}_l$, $l = 1,2,3$ are the canonical basis vectors. Our goal is to solve the boundary-value problem

$$\begin{aligned} G_0\mathbf{W} + G_x\partial_x\mathbf{W} + G_y\partial_y\mathbf{W} + G_z\partial_z\mathbf{W} = 0 \text{ in } \Omega, \\ (M_{\Gamma_m} - G_{\mathbf{n}})\mathbf{W} = 0 \text{ on } \Gamma_m \text{ and } (M_{\Gamma_a} - G_{\mathbf{n}})(\mathbf{W} - \mathbf{W}_{\text{inc}}) = 0 \text{ on } \Gamma_a, \end{aligned} \tag{2}$$

where $\mathbf{W}_{\text{inc}}$ is a given *incident field*, while M_{Γ_m} and M_{Γ_a} are trace operators defined on the *metallic* and *absorbing* boundaries Γ_m and Γ_a (see [10] for more details)

$$M_{\Gamma_m} = \begin{pmatrix} 0_{3\times 3} & N_{\mathbf{n}} \\ -N_{\mathbf{n}}^T & 0_{3\times 3} \end{pmatrix} \text{ and } M_{\Gamma_a} = |G_{\mathbf{n}}| = \begin{pmatrix} N_{\mathbf{n}}N_{\mathbf{n}}^T & 0_{3\times 3} \\ 0_{3\times 3} & N_{\mathbf{n}}^T N_{\mathbf{n}} \end{pmatrix}.$$

The matrices $G_{\mathbf{n}}^+$ and $G_{\mathbf{n}}^-$ are the positive and negative parts of $G_{\mathbf{n}}$ based on its diagonalization and we have that $|G_{\mathbf{n}}| = G_{\mathbf{n}}^+ - G_{\mathbf{n}}^-$.

2 Continuous classical and optimized Schwarz algorithms

We now decompose the domain Ω into two non-overlapping subdomains Ω_1 and Ω_2, and denote by Σ the interface between Ω_1 and Ω_2, by $\mathbf{W}_j$ the restriction of $\mathbf{W}$ to Ω_j and by $\mathbf{n}$ the unit outward normal vector to Σ directed from Ω_1 to Ω_2. Schwarz algorithms consist in computing iteratively $\mathbf{W}_j^{n+1}$ from $\mathbf{W}_j^n$, for $j = 1,2$

$$\begin{aligned} G_0\mathbf{W}_1^{n+1} + G_x\partial_x\mathbf{W}_1^{n+1} + G_y\partial_y\mathbf{W}_1^{n+1} + G_z\partial_z\mathbf{W}_1^{n+1} &= 0, \text{ in } \Omega_1, \\ (G_{\mathbf{n}}^- + S_1G_{\mathbf{n}}^+)\mathbf{W}_1^{n+1} &= (G_{\mathbf{n}}^- + S_1G_{\mathbf{n}}^+)\mathbf{W}_2^n, \text{ on } \Sigma, \\ G_0\mathbf{W}_2^{n+1} + G_x\partial_x\mathbf{W}_2^{n+1} + G_y\partial_y\mathbf{W}_2^{n+1} + G_z\partial_z\mathbf{W}_2^{n+1} &= 0, \text{ in } \Omega_2, \\ (G_{\mathbf{n}}^+ + S_2G_{\mathbf{n}}^-)\mathbf{W}_2^{n+1} &= (G_{\mathbf{n}}^+ + S_2G_{\mathbf{n}}^-)\mathbf{W}_1^n, \text{ on } \Sigma, \end{aligned} \tag{3}$$

where S_1 and S_2 are differential operators. When $S_1 = S_2 = 0_{6\times 6}$, the interface conditions become the positive and negative flux operators $G_{\mathbf{n}}^+$ and $G_{\mathbf{n}}^-$, and the *classical Schwarz algorithm* is obtained. Applying $G_{\mathbf{n}}^+$ (respectively $G_{\mathbf{n}}^-$) to a vector $\mathbf{W}$ means to select the characteristic variables associated to out-going (respectively in-coming) waves, which is very natural considering the hyperbolic nature of the problem, see [9] (section 3.1). We note that

Algorithm	1	2	3	4	5
$\mathscr{F}(\tilde{S}_j)$	0	$-\frac{s-i\omega}{s+i\omega}$	$-\frac{k^2+i\omega\sigma}{k^2-2\omega^2+i\omega\sigma+2i\omega s}$	$-\frac{s_j-i\omega}{s_j+i\omega}$	$-\frac{k^2+i\omega\sigma}{k^2-2\omega^2+i\omega\sigma+2i\omega s_j}$

Table 1 Five different choices for the symbols of the operators in the transmission conditions (6) leading to five different optimized Schwarz algorithms

$$\begin{aligned} G_{\mathbf{n}}^{-} &= \begin{pmatrix} -N_{\mathbf{n}}N_{\mathbf{n}}^T & N_{\mathbf{n}} \\ N_{\mathbf{n}}^T & -N_{\mathbf{n}}^T N_{\mathbf{n}} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_{3\times 3} \\ -N_{\mathbf{n}}^T \end{pmatrix} \begin{pmatrix} -N_{\mathbf{n}}N_{\mathbf{n}}^T & N_{\mathbf{n}} \end{pmatrix}, \\ G_{\mathbf{n}}^{+} &= \begin{pmatrix} N_{\mathbf{n}}N_{\mathbf{n}}^T & N_{\mathbf{n}} \\ N_{\mathbf{n}}^T & N_{\mathbf{n}}^T N_{\mathbf{n}} \end{pmatrix} = \begin{pmatrix} \mathbb{I}_{3\times 3} \\ N_{\mathbf{n}}^T \end{pmatrix} \begin{pmatrix} N_{\mathbf{n}}N_{\mathbf{n}}^T & N_{\mathbf{n}} \end{pmatrix}. \end{aligned} \tag{4}$$

Thus the classical transmission conditions are equivalent to impedance conditions,

$$\begin{aligned} G_{\mathbf{n}}^{-}\mathbf{W}_1^{n+1} = G_{\mathbf{n}}^{-}\mathbf{W}_2^{n} \Leftrightarrow \mathscr{B}_{\mathbf{n}}(\mathbf{E}_1^{n+1},\mathbf{H}_1^{n+1}) = \mathscr{B}_{\mathbf{n}}(\mathbf{E}_2^{n},\mathbf{H}_2^{n}), \\ G_{\mathbf{n}}^{+}\mathbf{W}_2^{n+1} = G_{\mathbf{n}}^{+}\mathbf{W}_1^{n} \Leftrightarrow \mathscr{B}_{-\mathbf{n}}(\mathbf{E}_2^{n+1},\mathbf{H}_2^{n+1}) = \mathscr{B}_{-\mathbf{n}}(\mathbf{E}_1^{n},\mathbf{H}_1^{n}). \end{aligned} \tag{5}$$

with $\mathscr{B}_{\mathbf{n}}(\mathbf{E},\mathbf{H}) = N_{\mathbf{n}}^T\mathbf{E} - N_{\mathbf{n}}^T N_{\mathbf{n}}\mathbf{H}$. For Ω_2 we have used the fact that $G_{\mathbf{n}}^{+} = -G_{-\mathbf{n}}^{-}$. The classical Schwarz algorithm is adopted in [10] together with low order DG methods in the 3D case. Along the lines of (5), we have the equivalences

$$\begin{aligned} &(G_{\mathbf{n}}^{-} + S_1 G_{\mathbf{n}}^{+})\mathbf{W}_1^{n+1} = (G_{\mathbf{n}}^{-} + S_1 G_{\mathbf{n}}^{+})\mathbf{W}_2^{n} \\ &\quad \Leftrightarrow (\mathscr{B}_{\mathbf{n}} + \tilde{S}_1\mathscr{B}_{-\mathbf{n}})(\mathbf{E}_1^{n+1},\mathbf{H}_1^{n+1}) = (\mathscr{B}_{\mathbf{n}} + \tilde{S}_1\mathscr{B}_{-\mathbf{n}})(\mathbf{E}_2^{n},\mathbf{H}_2^{n}), \\ &(G_{\mathbf{n}}^{+} + S_2 G_{\mathbf{n}}^{-})\mathbf{W}_2^{n+1} = (G_{\mathbf{n}}^{+} + S_2 G_{\mathbf{n}}^{-})\mathbf{W}_1^{n} \\ &\quad \Leftrightarrow (\mathscr{B}_{-\mathbf{n}} + \tilde{S}_2\mathscr{B}_{\mathbf{n}})(\mathbf{E}_2^{n+1},\mathbf{H}_2^{n+1}) = (\mathscr{B}_{-\mathbf{n}} + \tilde{S}_2\mathscr{B}_{\mathbf{n}})(\mathbf{E}_1^{n},\mathbf{H}_1^{n}), \end{aligned} \tag{6}$$

where $\tilde{S}_1$ and $\tilde{S}_2$ denote differential operators which are approximations of the transparent operators. From these transparent operators we can obtain a hierarchy of optimized algorithms with appropriate choices for $\tilde{S}_1$ and $\tilde{S}_2$ [11]. The operators S_1 and S_2 are eventually defined to guarantee the equivalences in (6).

If we consider the TM formulation of Maxwell's equations, that is with $\mathbf{E} = (\,0\;0\;E_z\,)^T$ and $\mathbf{H} = (\,H_x\;H_y\;0\,)^T$, then $\mathbf{W} = (\,E_z\;H_x\;H_y\,)^T$, $N_{\mathbf{n}} = (\,n_y\;-n_x\,)^T$, and

$$G_0 = \begin{pmatrix} \sigma + i\omega & 0_{1\times 2} \\ 0_{2\times 1} & i\omega\mathbb{I}_{2\times 2} \end{pmatrix}, \quad G_x = \begin{pmatrix} 0 & N_{\mathbf{e}_x} \\ N_{\mathbf{e}_x}^T & 0 \end{pmatrix} \quad \text{and} \quad G_y = \begin{pmatrix} 0 & N_{\mathbf{e}_y} \\ N_{\mathbf{e}_y}^T & 0 \end{pmatrix}.$$

We give in Table 1 the symbols $\mathscr{F}(\tilde{S}_j)$ of $\tilde{S}_j$ in the 2d case for conductive media for five different Schwarz algorithms, where the parameters $s = p(1+i)$, $s_1 = p_1(1+i)$ and $s_2 = p_2(1+i)$ are solutions of some min-max problems, as explained in [11] (section 5, table 5.1). Note that the Fourier symbols of the operators in algorithms 1, 2 and 4 are constants, therefore they have the same expression as in the physical space. In this case (6) can be written in the 2d situation considered here as

$$\begin{aligned} E_1^{n+1} - N_{\mathbf{n}}\mathbf{H}_1^{n+1} + \tilde{S}_1(E_1^{n+1} + N_{\mathbf{n}}\mathbf{H}_1^{n+1}) = E_2^{n} - N_{\mathbf{n}}\mathbf{H}_2^{n} + \tilde{S}_1(E_2^{n} + N_{\mathbf{n}}\mathbf{H}_2^{n}), \\ E_2^{n+1} + N_{\mathbf{n}}\mathbf{H}_2^{n+1} + \tilde{S}_2(E_2^{n+1} - N_{\mathbf{n}}\mathbf{H}_2^{n+1}) = E_1^{n} + N_{\mathbf{n}}\mathbf{H}_1^{n} + \tilde{S}_2(E_1^{n} - N_{\mathbf{n}}\mathbf{H}_1^{n}). \end{aligned} \tag{7}$$

This is not the case for algorithms 3 and 5 which involved second order transmission conditions. Here, the $\tilde{S}_j$ are operators whose Fourier symbols have the form

$$\mathscr{F}(\tilde{S}_j) = \frac{q_j(k)}{r_j(k)} \text{ with } q_j(k) = -(k^2 + i\omega\sigma) \text{ and } r_j(k) = k^2 - 2\omega^2 + i\omega\sigma + 2i\omega s_j.$$

where the Fourier variable k corresponds to a transform with respect to the tangential direction τ along the interface, assuming a two-subdomain decomposition with a straight interface. In that case, $\mathscr{F}^{-1}(q_j)$ and $\mathscr{F}^{-1}(r_j)$ are partial differential operators in the τ variable,

$$\mathscr{F}^{-1}(q_j) = \partial_{\tau\tau} - i\omega\sigma, \mathscr{F}^{-1}(r_j) = -\partial_{\tau\tau} - 2\omega^2 + i\omega\sigma + 2i\omega s_j, s_j \in \mathbb{C},$$

and (7) can be re-written as

$$\begin{aligned}
\mathscr{F}^{-1}\left(r_1(E_1^{n+1} - N_{\mathbf{n}}\mathbf{H}_1^{n+1})\right) &+ \mathscr{F}^{-1}\left(q_1(E_1^{n+1} + N_{\mathbf{n}}\mathbf{H}_1^{n+1})\right) \\
&= \mathscr{F}^{-1}\left(r_1(E_2^{n} - N_{\mathbf{n}}\mathbf{H}_2^{n})\right) + \mathscr{F}^{-1}\left(q_1(E_2^{n} + N_{\mathbf{n}}\mathbf{H}_2^{n})\right), \\
\mathscr{F}^{-1}\left(r_2(E_2^{n+1} + N_{\mathbf{n}}\mathbf{H}_2^{n+1})\right) &+ \mathscr{F}^{-1}\left(q_2(E_2^{n+1} - N_{\mathbf{n}}\mathbf{H}_2^{n+1})\right) \\
&= \mathscr{F}^{-1}\left(r_2(E_1^{n} + N_{\mathbf{n}}\mathbf{H}_1^{n})\right) + \mathscr{F}^{-1}\left(q_2(E_1^{n} - N_{\mathbf{n}}\mathbf{H}_1^{n})\right).
\end{aligned}$$

3 Discontinuous Galerkin approximation

Let $\mathscr{T}_h$ be a discretization of Ω and Γ^0, Γ^m and Γ^a be the sets of purely internal, metallic and absorbing faces of $\mathscr{T}_h$. We denote by K an element of $\mathscr{T}_h$ and by $F = K \cap \tilde{K}$ the face shared by two neighboring elements K and $\tilde{K}$. On this face F, we define the *average* by $\{\mathbf{W}\} = \frac{1}{2}(\mathbf{W}_K + \mathbf{W}_{\tilde{K}})$ and the *tangential trace jump* by $[[\mathbf{W}]] = G_{\mathbf{n}_K}\mathbf{W}_K + G_{\mathbf{n}_{\tilde{K}}}\mathbf{W}_{\tilde{K}}$. For two vector functions $\mathbf{U}$ and $\mathbf{V}$ in $(L^2(D))^6$, we denote $(\mathbf{U},\mathbf{V})_D = \int_D \mathbf{U}\cdot\overline{\mathbf{V}}\,\mathrm{d}x$, if D is a domain of $\mathbb{R}^3$ and $\langle\mathbf{U},\mathbf{V}\rangle_F = \int_F \mathbf{U}\cdot\overline{\mathbf{V}}\,\mathrm{d}s$ if F is a face of $\mathbb{R}^2$. For sake of simplicity, we will skip some subscripts, that is $(\cdot,\cdot) = (\cdot,\cdot)_{\mathscr{T}_h} = \sum_{K\in\mathscr{T}_h}(\cdot,\cdot)_K$. On the boundaries we define

$$M_{F,K} = \begin{cases} \begin{pmatrix} \eta_F N_{\mathbf{n}_K} N_{\mathbf{n}_K}^T & N_{\mathbf{n}_K} \\ -N_{\mathbf{n}_K}^T & 0_{3\times 3} \end{pmatrix} \text{ with } \eta_F \neq 0, \text{ if } F \text{ belongs to } \Gamma^m, \\ |G_{\mathbf{n}_K}| \text{ if } F \text{ belongs to } \Gamma^a. \end{cases}$$

Using these notations, the weak formulation of the problem is

$$\begin{aligned}
(G_0\mathbf{W},\mathbf{V}) + \left(\sum_{l\in\{x,y,z\}} G_l\partial_l\mathbf{W},\mathbf{V}\right) - \sum_{F\in\Gamma^0}\langle[[\mathbf{W}]],\{\mathbf{V}\}\rangle_F + \sum_{F\in\Gamma^0}\left\langle\frac{1}{2}[[\mathbf{W}]],\{\mathbf{V}\}\right\rangle_F \\
+ \sum_{F\in\Gamma^m\cup\Gamma^a}\left\langle\frac{1}{2}(M_{F,K} - G_{\mathbf{n}_K})\mathbf{W},\mathbf{V}\right\rangle_F = \sum_{F\in\Gamma^a}\left\langle\frac{1}{2}(M_{F,K} - G_{\mathbf{n}_K})\mathbf{W}_{\text{inc}},\mathbf{V}\right\rangle_F.
\end{aligned}$$

Note that we have implicitly adopted an upwind scheme for the calculation of the boundary integral over an internal face $F \in \Gamma^0$. An alternative choice is that of a centered scheme. Both of these options are discussed and compared in [8]. Let $\mathbb{P}_p(D)$ denote the space of polynomial functions of degree at most p on a domain D. For any element $K \in \mathscr{T}_h$, let $\mathbf{D}^p(K) \equiv (\mathbb{P}_p(K))^6$. The vectors $\mathbf{W}$ and $\mathbf{V}$ will be taken in the space $\mathbf{D}_h^p = \left\{\mathbf{V} \in (L^2(\Omega))^6 \mid \mathbf{V}|_K \in \mathbf{D}^p(K),\ \forall K \in \mathscr{T}_h\right\}$.

For the discretization of optimized transmission conditions, let Γ_Σ be the set of faces on Σ, Γ_0^j be the set of interior faces of Ω_j and Γ_b^j be the set of faces of Ω_j lying on $\partial\Omega$. Then the weak form in the two-subdomain case can be written as

$$\begin{aligned}
&\mathscr{L}(\mathbf{W}_1,\mathbf{V}_1) + \sum_{\Gamma_0^1} \diamond + \sum_{\Gamma_b^1} \diamond + \sum_{F\in\Gamma_\Sigma} \left\langle \frac{1}{2}\left(|G_{\mathbf{n}_K}| - G_{\mathbf{n}_K}\right)(\mathbf{W}_1 - \mathbf{W}_2), \mathbf{V}_1 \right\rangle_F = 0,\\
&\mathscr{L}(\mathbf{W}_2,\mathbf{V}_2) + \sum_{\Gamma_0^2} \diamond + \sum_{\Gamma_b^2} \diamond + \sum_{F\in\Gamma_\Sigma} \left\langle \frac{1}{2}\left(|G_{\mathbf{n}_{\tilde K}}| - G_{\mathbf{n}_{\tilde K}}\right)(\mathbf{W}_2 - \mathbf{W}_1), \mathbf{V}_2 \right\rangle_F = 0,
\end{aligned} \tag{8}$$

where $\mathscr{L}(\mathbf{W}_j,\mathbf{V}_j) \equiv (G_0\mathbf{W}_j,\mathbf{V}_j) + (\sum_l G_l\partial_l\mathbf{W}_j,\mathbf{V}_j)$ and, for simplicity, we have replaced some terms on the faces that are not important for the presentation by a $\diamond$. For any face $F = K\cap\tilde K$ on Σ, if $\mathbf{n}$ denotes the normal on Σ directed from Ω_1 towards Ω_2, and K and $\tilde K$ are elements of Ω_1 and Ω_2, we have $\mathbf{n}_K = \mathbf{n} = -\mathbf{n}_{\tilde K}$. In order to simplify the notation, we make use of $G_{\mathbf{n}}^- = \frac{1}{2}(G_{\mathbf{n}} - |G_{\mathbf{n}}|)$ and $G_{\mathbf{n}}^+ = \frac{1}{2}(G_{\mathbf{n}} + |G_{\mathbf{n}}|)$. Then, starting from initial guesses $\mathbf{W}_1^0$ and $\mathbf{W}_2^0$, the classical Schwarz algorithm computes the iterates $\mathbf{W}_j^{n+1}$ from $\mathbf{W}_j^n$ by solving on Ω_1 and Ω_2 the subproblems

$$\begin{aligned}
&\mathscr{L}(\mathbf{W}_1^{n+1},\mathbf{V}_1) + \sum_{\Gamma_0^1} \diamond + \sum_{\Gamma_b^1} \diamond - \sum_{F\in\Gamma_\Sigma} \left\langle G_{\mathbf{n}}^-(\mathbf{W}_1^{n+1} - \mathbf{W}_2^n), \mathbf{V}_1 \right\rangle_F = 0,\\
&\mathscr{L}(\mathbf{W}_2^{n+1},\mathbf{V}_2) + \sum_{\Gamma_0^2} \diamond + \sum_{\Gamma_b^2} \diamond + \sum_{F\in\Gamma_\Sigma} \left\langle G_{\mathbf{n}}^+(\mathbf{W}_2^{n+1} - \mathbf{W}_1^n), \mathbf{V}_2 \right\rangle_F = 0.
\end{aligned} \tag{9}$$

In order to introduce optimized transmission conditions (3) into the DG discretization, we first want to show explicitly what transmission conditions the classical relaxation in (9) corresponds to. To do so, the subdomain problems solved in (9) are not allowed to depend on variables of the other subdomain anymore, since the coupling will be performed with the transmission conditions, and we thus need to introduce additional unknowns, namely $\mathbf{W}_{2,\Omega_1}^{n+1}$ on Ω_1 and $\mathbf{W}_{1,\Omega_2}^{n+1}$ on Ω_2, in order to write the classical Schwarz iteration with local variables only, *i.e.*

$$\begin{aligned}
&\mathscr{L}(\mathbf{W}_1^{n+1},\mathbf{V}_1) + \sum_{\Gamma_0^1} \diamond + \sum_{\Gamma_b^1} \diamond - \sum_{F\in\Gamma_\Sigma} \left\langle G_{\mathbf{n}}^-(\mathbf{W}_1^{n+1} - \mathbf{W}_{2,\Omega_1}^{n+1}), \mathbf{V}_1 \right\rangle_F = 0,\\
&\mathscr{L}(\mathbf{W}_2^{n+1},\mathbf{V}_2) + \sum_{\Gamma_0^2} \diamond + \sum_{\Gamma_b^2} \diamond + \sum_{F\in\Gamma_\Sigma} \left\langle G_{\mathbf{n}}^+(\mathbf{W}_2^{n+1} - \mathbf{W}_{1,\Omega_2}^{n+1}), \mathbf{V}_2 \right\rangle_F = 0.
\end{aligned} \tag{10}$$

Comparing with the classical Schwarz algorithm (9), we see that in order to obtain the same algorithm, the transmission conditions for (10) need to be chosen as $G_{\mathbf{n}}^-\mathbf{W}_{2,\Omega_1}^{n+1} = G_{\mathbf{n}}^-\mathbf{W}_2^n$ and $G_{\mathbf{n}}^+\mathbf{W}_{1,\Omega_2}^{n+1} = G_{\mathbf{n}}^+\mathbf{W}_1^n$, which implies that at the limit, when

the algorithm converges, we must verify the coupling conditions

$$G_{\mathbf{n}}^{-}\mathbf{W}_{2,\Omega_1} = G_{\mathbf{n}}^{-}\mathbf{W}_2, \qquad G_{\mathbf{n}}^{+}\mathbf{W}_{1,\Omega_2} = G_{\mathbf{n}}^{+}\mathbf{W}_1, \tag{11}$$

where we dropped the iteration index to denote the limit quantities. The Schwarz algorithm (10) can however also be used with optimized transmission conditions (3), which have to be the DG discretization of the strong relations

$$\begin{array}{l} G_{\mathbf{n}}^{-}\mathbf{W}_{2,\Omega_1}^{n+1} + S_1 G_{\mathbf{n}}^{+}\mathbf{W}_1^{n+1} = G_{\mathbf{n}}^{-}\mathbf{W}_2^{n} + S_1 G_{\mathbf{n}}^{+}\mathbf{W}_{1,\Omega_2}^{n}, \\ G_{\mathbf{n}}^{+}\mathbf{W}_{1,\Omega_2}^{n+1} + S_2 G_{\mathbf{n}}^{-}\mathbf{W}_2^{n+1} = G_{\mathbf{n}}^{+}\mathbf{W}_1^{n} + S_2 G_{\mathbf{n}}^{-}\mathbf{W}_{2,\Omega_1}^{n}. \end{array} \tag{12}$$

Then, we want to show the equivalence between (11) and the DG discretization we adopt for the transmission conditions (12) at convergence in a 2d case. First, from (4) note that relation (11) is equivalent to

$$\begin{array}{l} N_{\mathbf{n}}N_{\mathbf{n}}^{T}\mathbf{E}_{2,\Omega_1} - N_{\mathbf{n}}\mathbf{H}_{2,\Omega_1} = N_{\mathbf{n}}N_{\mathbf{n}}^{T}\mathbf{E}_2 - N_{\mathbf{n}}\mathbf{H}_2, \\ N_{\mathbf{n}}N_{\mathbf{n}}^{T}\mathbf{E}_{1,\Omega_2} + N_{\mathbf{n}}\mathbf{H}_{1,\Omega_2} = N_{\mathbf{n}}N_{\mathbf{n}}^{T}\mathbf{E}_1 + N_{\mathbf{n}}\mathbf{H}_1. \end{array} \tag{13}$$

We translate these relations using auxiliary variables $\Lambda_{2,\Omega_1} := E_{2,\Omega_1} - N_{\mathbf{n}}\mathbf{H}_{2,\Omega_1}$, $\Lambda_2 := E_2 - N_{\mathbf{n}}\mathbf{H}_2$, $\Lambda_{1,\Omega_2} := E_{1,\Omega_2} + N_{\mathbf{n}}\mathbf{H}_{1,\Omega_2}$ and $\Lambda_1 := E_1 + N_{\mathbf{n}}\mathbf{H}_1$ belonging to the trace space $M_h^p = \left\{\eta \in L^2(\Sigma) \mid \eta|_F \in \mathbb{P}_p(F), \forall F \in \Sigma\right\}$. Then (13) becomes

$$\Lambda_{2,\Omega_1} = \Lambda_2 \ \text{ and } \ \Lambda_{1,\Omega_2} = \Lambda_1. \tag{14}$$

From (12) and (14), we have to find for optimized transmission conditions a suitable DG discretization of the relations

$$\Lambda_{2,\Omega_1} + \tilde{S}_1\Lambda_1 = \Lambda_2 + \tilde{S}_1\Lambda_{1,\Omega_2} \ \text{ and } \ \Lambda_{1,\Omega_2} + \tilde{S}_2\Lambda_2 = \Lambda_1 + \tilde{S}_2\Lambda_{2,\Omega_1}. \tag{15}$$

We focus on the case of second order transmission conditions and (15) becomes

$$\begin{array}{l} (-\partial_\tau^2 + i\omega\sigma - 2\omega^2 + 2i\omega s_1)(\Lambda_{2,\Omega_1} - \Lambda_2) + (-\partial_\tau^2 + i\omega\sigma)(\Lambda_{1,\Omega_2} - \Lambda_1) = 0, \\ (-\partial_\tau^2 + i\omega\sigma - 2\omega^2 + 2i\omega s_2)(\Lambda_{1,\Omega_2} - \Lambda_1) + (-\partial_\tau^2 + i\omega\sigma)(\Lambda_{2,\Omega_1} - \Lambda_2) = 0. \end{array} \tag{16}$$

Let $(\eta_j)_j$ be a basis of M_h^p. We define the discrete matrices M_Σ and K_Σ by

$$\begin{aligned} (M_\Sigma)_{i,j} &= \sum_{F\in\Sigma} \langle \eta_i, \eta_j \rangle_F, \\ (K_\Sigma)_{i,j} &= \sum_{F\in\Sigma} \langle \partial_\tau\eta_i, \partial_\tau\eta_j \rangle_F + \sum_{n\in\Sigma^0} \alpha_n h^{-1} [[[[\eta_i]]]]_n [[[[\eta_j]]]]_n \\ &\quad - \sum_{n\in\Sigma^0} \{\{\partial_\tau\eta_i\}\}_n [[[[\eta_j]]]]_n - [[[[\eta_i]]]]_n \left\{\left\{\partial_\tau\eta_j\right\}\right\}_n, \end{aligned}$$

where positiveness is guaranteed for sufficiently large α_n, Σ^0 denotes the set of interior nodes of Σ, $[[[[\cdot]]]]_n$ and $\{\{\cdot\}\}_n$ denotes the jump and the average at a node n between values of the neighboring segments. The matrix K_Σ comes from the discretization of $-\partial_\tau^2$ using a symmetric interior penalty approach [2]. If we denote by $A_\Sigma = (K_\Sigma + i\omega\sigma M_\Sigma)$, the DG discretization of (16) we consider is

$$\begin{pmatrix} A_\Sigma - 2(\omega^2 - i\omega s_1)M_\Sigma & A_\Sigma \\ A_\Sigma & A_\Sigma - 2(\omega^2 - i\omega s_2)M_\Sigma \end{pmatrix} \begin{pmatrix} \Lambda_{2,\Omega_1} - \Lambda_2 \\ \Lambda_{1,\Omega_2} - \Lambda_1 \end{pmatrix} = 0. \quad (17)$$

Theorem 1. *If s_1 and s_2 are defined as given in [11] (section 5, table 5.1) then relations (14) and (17) are equivalent.*

The proof is based on the invertibility of the matrix of (17) and can be found in [7].

4 Numerical results

In order to illustrate numerically the proposed discrete versions of the optimized Schwarz algorithms, we consider the propagation of a plane wave in a homogeneous conductive medium with $\Omega = [0,1]^2$ and $\sigma = 0.5$. We use DG with several orders of polynomial interpolation, denoted by DG-$\mathbb{P}_k$ with $k = 1,2,3,4$, and impose on $\partial\Omega = \Gamma_a$ an incident wave $\mathbf{W}_{\text{inc}} = (\ \frac{k_y}{\omega}\ \ \frac{-k_x}{\omega}\ \ 1\ )^T e^{-i\mathbf{k}\cdot\mathbf{x}}$, and $\mathbf{k} = (\ k_x\ k_y\ )^T = (\ \omega\sqrt{1 - i\frac{\sigma}{\omega}}\ \ 0\ )^T$. The domain Ω is decomposed into two subdomains $\Omega_1 = [0,0.5] \times [0,1]$ and $\Omega_2 = [0.5,1] \times [0,1]$. The aim is to retrieve numerically the asymptotic behavior of the convergence factors of the optimized Schwarz methods. It has been proved that these factors behave like $1 - O(h^{\alpha_i})$, $i = 2,3,4,5$. We show here that numerically they behave like $1 - O(h^{\beta_i})$, $i = 2,3,4,5$, with $\beta_i \approx \alpha_i$. The performance of these algorithms is summarized in Figure 1.

References

1. Alonso-Rodriguez, A., Gerardo-Giorda, L.: New nonoverlapping domain decomposition methods for the harmonic Maxwell system. SIAM J. Sci. Comput. **28**(1), 102–122 (2006)
2. Arnold, D., Brezzi, F., Cockburn, B., Marini, L.: Unified analysis of Discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal. **39**(5), 1749–1779 (2002)
3. Chevalier, P., Nataf, F.: An OO2 (Optimized Order 2) method for the Helmholtz and Maxwell equations. In: 10th International Conference on Domain Decomposition Methods in Science and in Engineering, pp. 400–407. AMS, Boulder, Colorado, USA (1997)
4. Collino, P., Delbue, G., Joly, P., Piacentini, A.: A new interface condition in the non-overlapping domain decomposition for the Maxwell equations. Comput. Methods Appl. Mech. Engrg. **148**, 195–207 (1997)
5. Després, B.: Décomposition de domaine et problème de Helmholtz. C.R. Acad. Sci. Paris **1**(6), 313–316 (1990)
6. Després, B., Joly, P., Roberts, J.: A domain decomposition method for the harmonic Maxwell equations. In: Iterative methods in linear algebra, pp. 475–484. North-Holland, Amsterdam (1992)
7. Dolean, V., Bouajaji, M.E., Gander, M.J., Lanteri, S., Perrussel, R.: Discontinuous Galerkin discretizations of Optimized Schwarz methods for solving the time-harmonic Maxwell equations. in preparation (2014)
8. Dolean, V., Fol, H., Lanteri, S., Perrussel, R.: Solution of the time-harmonic Maxwell equations using discontinuous Galerkin methods. J. Comp. Appl. Math. **218**(2), 435–445 (2008)

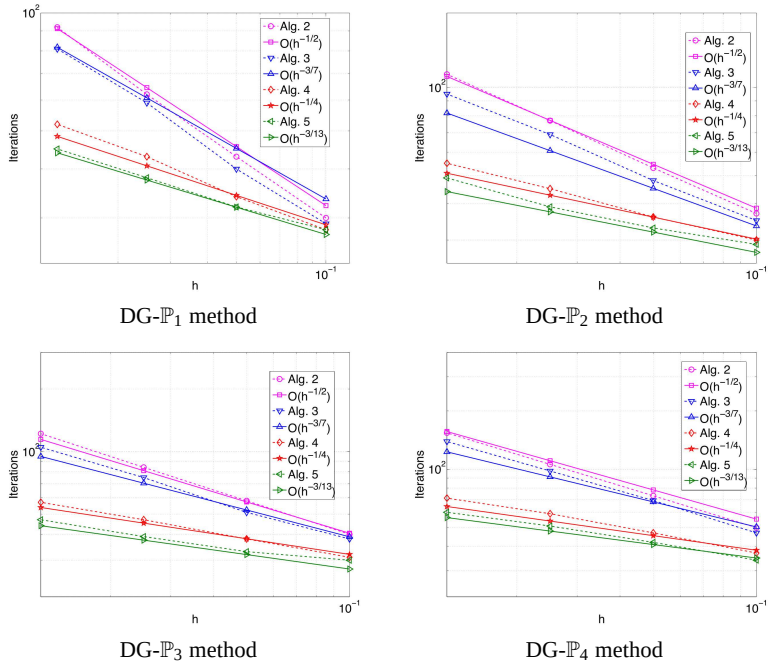


Fig. 1 Wave propagation in a homogeneous medium. Iteration count vs. h.

9. Dolean, V., Gerardo-Giorda, L., Gander, M.J.: Optimized Schwarz methods for Maxwell equations. SIAM J. Scient. Comp. **31**(3), 2193–2213 (2009)
10. Dolean, V., Lanteri, S., Perrussel, R.: A domain decomposition method for solving the three-dimensional time-harmonic Maxwell equations discretized by discontinuous Galerkin methods. J. Comput. Phys. **227**(3), 2044–2072 (2008)
11. El Bouajaji, M., Dolean, V., Gander, M.J., Lanteri, S.: Optimized Schwarz methods for the time-harmonic Maxwell equations with dampimg. SIAM J. Scient. Comp. **34**(4), 2048–2071 (2012)
12. Gander, M.J., Magoulès, F., Nataf, F.: Optimized Schwarz methods without overlap for the Helmholtz equation. SIAM J. Sci. Comput. **24**(1), 38–60 (2002)
13. Hesthaven, J., Warburton, T.: Nodal Discontinuous Galerkin methods: algorithms, analysis and applications. Springer (2008)
14. Lee, S.C., Vouvakis, M., Lee, J.F.: A non-overlapping domain decomposition method with non-matching grids for modeling large finite antenna arrays. J. Comput. Phys. **203**(1), 1–21 (2005)
15. Peng, Z., Lee, J.F.: Non-conformal domain decomposition method with second-order transmission conditions for time-harmonic electromagnetics. J. Comput. Phys. **229**(16), 5615–5629 (2010)
16. Peng, Z., Rawat, V., Lee, J.F.: One way domain decomposition method with second order transmission conditions for solving electromagnetic wave problems. J. Comput. Phys. **229**(4), 1181–1197 (2010)
17. Rawat, V., Lee, J.F.: Nonoverlapping domain decomposition with second order transmission condition for the time-harmonic Maxwell's equations. SIAM J. Sci. Comput. **32**(6), 3584–3603 (2010)

Simulations of micro channel gas flows with domain decomposition technique for kinetic and fluid dynamics equations

Sudarshan Tiwari[1], Axel Klar[1] and Steffen Hardt[2]

1 Introduction

In the last 20 years many research papers have been reported about the development of domain decompositions for the kinetic and the fluid dynamic equations, see for example [7, 8, 10, 14, 11, 12, 15, 16]. From large to small scale geometries one may experience different degrees of rarefaction of a gas. The degrees of rarefaction of a gas can be measured by the Knudsen number $Kn = \lambda/L$, where λ is the mean free path and L is the characteristic length, for example the channel width. For $Kn < 0.001$, the flow is in the continuum regime, the compressible Navier-Stokes equations with no-slip boundary conditions are solved. For $0.001 < Kn < 0.1$, the flow is in the slip regime, where the Navier-Stokes equations with velocity-slip and temperature jump conditions are solved [1]. For $Kn > 0.1$ a kinetic type approach, based on the Boltzmann equation is required. We note that the kinetic approach is valid in the whole range of rarefaction of a gas. At standard conditions the mean free path of a gas in a micro- or nano channel is of the order L or larger, so the Knudsen number is no longer small. Therefore, the fluid dynamic equations, the compressible Euler or Navier-Stokes equations, cannot predict the flows correctly in a small scale geometry [9].

In this paper we present stationary solutions of a Poiseuille flow in a micro channel. We have considered the large range of Knudsen numbers. We use the domain decomposition of the Boltzmann and the compressible Navier-Stokes equations. We have coupled a meshfree particle method for the compressible Navier-Stokes equations and a DSMC type of particle method for the Boltzmann equation. We have first observed the discrepancy in the Boltzmann and Navier-Stokes solutions. Then we have defined boundary layers and solved the Boltzmann equations in the boundary layers and the Navier-Stokes equations in the rest of the channel. We have used the standard interface boundary conditions between both domains, see [16, 15]. Alternatively, we have solved the Navier-Stokes equations until steady state has been reached. It gives quite diffusive solutions, however, this is the good candidate to initialize the Boltzmann solver. One can apply a breakdown criterion to the stationary Navier-Stokes equations and then decompose the Boltzmann and Navier-Stokes domains.

[1] Department of Mathematics, TU Kaiserslautern, Erwin-Schroendinger Strasse, 67663 Kaiserslautern, Germany e-mail: {tiwari}{klar}@mathematik.uni-kl.de .[2] Center of Smart Interfaces, TU Darmstadt, Petersenstr. 32, 64287 Darmstadt, Germany e-mail: hardt@csi.tu-darmstadt.de

The paper is organized as follows. In section 2 we present the mathematical models and numerical methods. In section 3 we discuss the numerical solutions and the domain decompositions.

2 Governing equations and numerical methods

In this section we introduce the Boltzmann equation, the Navier-Stokes equations as its hydrodynamic limit, numerical methods and domain decomposition strategies.

2.1 The Boltzmann equation and its hydrodynamic limits

The Boltzmann equation describes the time evolution of a distribution function $f(t,x,v)$ for particles of velocity $v \in \mathrm{Re}^3$ at $x \in D \subset \mathrm{Re}^s (s = 1,2,3)$ and time $t \in \mathrm{Re}_+$. It is given by

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f = Q(f,f), \tag{1}$$

where

$$Q(f,f) = \int_{\mathrm{Re}^3} \int_{S^2} \beta(|v-w|,\eta)[f(v')f(w') - f(v)f(w)]d\omega(\eta)dw$$

with

$$v' = T_{v,w}(\eta) = v - \eta < \eta, v-w >,\ w' = T_{w,v}(\eta).$$

Here, β denotes the collision cross section, η is the unit normal vector on the sphere, $d\omega(\eta)$ is the solid-angle element in the direction of η and $<,>$ is the scalar product. For the sake of simplicity, we have not used any bold letters for vector quantities, like x, v, w, etc. Writing the equations in dimensionless form one observes that Q is of the order $\mathscr{O}(\frac{1}{Kn})$. The local mean free path $\lambda = \lambda(x,t)$ is given by

$$\lambda = \frac{kT}{\sqrt{2}\pi p d^2}, \tag{2}$$

where k is the Boltzmann constant, $T = T(x,t)$ the temperature, $p = p(x,t)$ the pressure and d is the diameter of molecules. For more details we refer to [6]. For Kn tending to zero one can show that the Boltzmann distribution function f tends to the local Maxwellian [5]

$$f_M(t,x,v) = \frac{\rho}{(2\pi RT)^{3/2}}\, e^{-\frac{|v-U|^2}{2RT}}, \tag{3}$$

where $\rho = \rho(x,t)$ is the density, $U = U(x,t)$ the mean velocity and R is the gas constant. The parameters of the Maxwellian ρ, U, T solve the compressible Euler equations. This can be verified from the asymptotic expansion of f in Kn, where the zeroth order approximation gives the local Maxwellian distribution and the first order approximation [3] gives the Chapman-Enskog distribution

$$f_{CE}(t,x,v) = f_M(t,x,v)\,[1+\phi(t,x,v)]\,, \tag{4}$$

with

$$\phi(t,x,v) = \frac{2}{5}\frac{q\cdot c}{\rho(RT)^2}\left(\frac{|c|^2}{2RT}-\frac{5}{2}\right) - \frac{1}{2}\frac{\tau : c\otimes c}{\rho(RT)^2}, \tag{5}$$

where $c = v - U$. Here, $\phi = \mathscr{O}(Kn)$ and the parameters ρ, U, T, q, τ satisfy the compressible Navier-Stokes equations

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla\cdot(\rho U) &= 0 \\ \frac{\partial(\rho U)}{\partial t} + \nabla\cdot(\rho U\otimes U + pI - \tau) &= 0 \\ \frac{\partial(\rho E)}{\partial t} + \nabla\cdot[(\rho E + p)U - \tau\cdot U - q] &= 0, \end{aligned} \tag{6}$$

where $E = |U|^2/2 + e$ is the total energy and e is the internal energy. The stress tensor τ and heat flux vector q are of order Kn and given by

$$\tau_{ij} = \mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{2}{3}\nabla\cdot U\ \delta_{ij}\right), \qquad q = -\kappa\nabla T. \tag{7}$$

The dynamic viscosity $\mu = \mu(x,t)$ and the heat conductivity $\kappa = \kappa(x,t)$ for a monatomic gas of hard sphere molecules are of order Kn. They are given, see [4], by

$$\mu = \frac{5}{16d^2}\sqrt{\frac{mkT}{\pi}}, \qquad \kappa = \frac{15k}{4m}\mu, \tag{8}$$

where m is the molecular mass. In this paper we have considered a monatomic gas of hard spheres.

2.2 Numerical methods

We apply Lagrangian particle methods of different characters for both types of equations. The Boltzmann equation is solved by a DSMC type Monte Carlo method, whereas the Navier-Stokes equations are treated with a meshfree particle method, which is called the Finite Pointset Method (FPM).

2.2.1 Particle Method for the Boltzmann equation

For solving the Boltzmann equation we have used a variant of the DSMC method [4], developed in [13, 2]. The method is based on the time splitting of the Boltzmann equation. Introducing fractional steps one solves first the free transport equation (the collisionless Boltzmann equation) for one time step. During the free flow, boundary and interface conditions are taken into account. In a second step (the collision step) the spatially homogeneous Boltzmann equation without the transport term is solved. To simulate this equation by a particle method an explicit Euler step is performed. The result is then used in the next time step as the new initial condition for the free flow. To solve the homogeneous Boltzmann equation the key point is to find an efficient particle approximation of the product distribution functions in the Boltzmann collision operator given only an approximation of the distribution function itself. To guarantee positivity of the distribution function during the collision step a restriction of the time step proportional to the Knudsen number is needed. That means that the method becomes exceedingly expensive for small Knudsen numbers.

2.2.2 Meshfree particle method for the Navier-Stokes equations

We solve the Navier-Stokes equations by a meshfree Lagrangian particle method. We approximate the spatial derivatives at an arbitrary particle from its surrounding clouds of points with the help of the least squares method. We express the compressible Navier-Stokes equations in primitive variables according to the Lagrangian form. We first fill a computational domain by a finite number of particles and assign all fluid quantities to them. Then we approximate the spatial derivatives at every particle position. The resulting equations reduce to a time dependent system of ordinary differential equations. This system can be solved by a simple integration scheme. One can use the explicit Euler scheme, but this requires a very small time step. Here a two step Runge-Kutta method is used which is sufficient for the test cases considered in this paper. Due to space limitations, we do not present the meshfree method, we refer to our earlier reports, see [17, 16].

2.2.3 Coupling particle methods for the Boltzmann and the compressible Navier-Stokes equations

The DSMC method is a mesh-based method since gas molecules have to be sorted into cells for the intermolecular collisions. As already described, the compressible Navier-Stokes equations are solved by a meshfree method. Therefore, we need to couple the mesh-based and the meshfree particle methods. We decompose a domain into Boltzmann and Navier-Stokes domains, then we have to prescribe the interface boundary conditions from one domain into another domain.

In order to apply the interface boundary conditions for the Boltzmann equation, we have to define the boundary cells (or interface cells) in the Navier-Stokes do-

main. On these buffer cells we generate gas molecules according to a Maxwellian distribution, where the parameters are approximated from the Navier-Stokes equations. If the gas molecules leave the Boltzmann domain and enter to Navier-Stokes one, we delete them.

The interface boundary conditions for the Navier-Stokes equations are applied as follows. In the Boltzmann domain we sample and store the macroscopic quantities at the cell centers. Near the interface there may be several Boltzmann cell centers, which are the neighbor of a Navier-Stokes particle. In this case we consider all neighboring Boltzmann cells and approximate the spatial derivatives from the least squares method. Instead of using the Dirichlet boundary condition at the Boltzmann interface cell, we find this approach is sufficient. When the Navier-Stokes particles leave the Navier-Stokes domain, we delete them. If they thinned out the domain, we add new particles and interpolate the data from its neighboring particle values.

It is well known that in all DSMC type solvers there are some statistical fluctuations in the solutions of the Boltzmann equation. These fluctuating data destabilize the Navier-Stokes solver. Therefore, we need a smoothing operator, see [16, 15] for details.

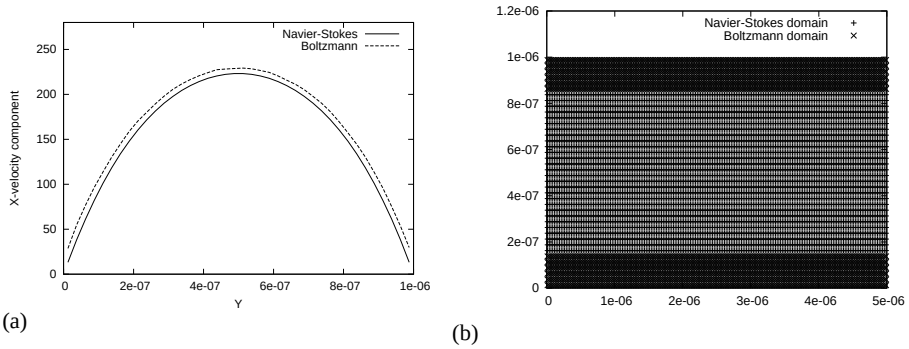
3 Numerical results

We consider a micro channel of size $[0,\ 5\cdot H]\times[0,\ H]$ with $H = 1\cdot 10^{-6}m$ as shown in Fig. 1(b). The left and right walls are inflow and outflow boundaries, respectively and the upper and lower are solid wall boundaries. While solving the Navier-Stokes equations, we prescribe a temperature T_{in} and a pressure p_{in} on the inflow boundary. Similarly, we prescribe a pressure p_{out} on the outflow boundary. We use the Neumann boundary conditions for the velocity and temperature, on the in- and outflow boundaries. Furthermore, zero velocity and $T = T_0$ are considered on the upper and lower boundaries, where T_0 is the initial temperature of the gas. We choose Argon as a gas with a molecular mass $m = 6.63\cdot 10^{-26}kg$. The Boltzmann constant $k = 1.38\cdot 10^{-23}JK^{-1}$, the molecular diameter $d = 3.68\cdot 10^{-10}m$, the ratio of specific heats $\gamma = 5/3$ enter as parameters. These parameters give the gas constant $R = 208JkgK^{-1}$. The dynamic viscosity and thermal conductivity in the compressible Navier-Stokes equations are assumed to be constant and are evaluated with the initial temperature according to eq. (8). The initial velocity is zero. The initial pressure is $(p_{in}+p_{out})/2$ and the initial density is determined from the ideal gas law.

When we solve the Boltzmann equation we initialize the gas according to the Maxwellian distribution in each cell with the initial parameters as described for the Navier-Stokes solver. We generate the molecules according to the Maxwellian distribution at the inflow boundary, where the density is determined from the given pressure and the temperature using the ideal gas law. The mean velocity is extrapolated from the the interior of the Boltzmann cells. Similarly, we also generate the molecules according to the Maxwellian distribution at the outflow boundary, where we extrapolate the mean velocity and the temperature from the interior cell values

and the pressure is given. If the molecules leave the inflow or outflow boundary we delete them. On the upper and lower walls we use the diffuse reflection with thermal accommodation. We choose 200×40 cells for the Boltzmann solver and the mesh-free particles for the Navier-Stokes solver of the same order. For the Navier-Stokes solver we choose the time step Δt equal to $3 \cdot 10^{-11} s$ and $0.5 * \Delta x / \sqrt{(2RT0)}$, where Δx is the cell size. In all cases we compute upto the final time $t = 1 \cdot 10^{-6} s$.

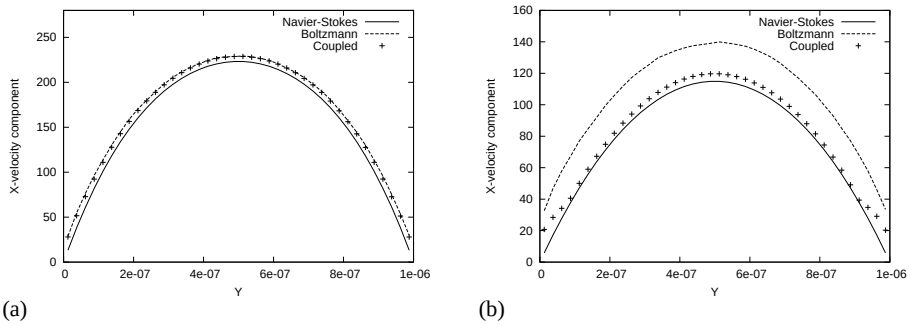
In the first test case, we consider $p_{in} = 624000Pa, p_{out} = 208000Pa$ and $T_0 = 300K$. This gives the Knudsen number on the left of 0.01101 and on the right of 0.03303. We are now in the slip regime, where we expect the Navier-Stokes solutions with no slip boundary conditions do not match with the Boltzmann ones. In Fig. 1(a) the x component of velocities from both solvers at 2/3rd of the channel length along the y axis are plotted. We observe that there is a discrepancy between the solutions of both equations. It is required to use slip boundary conditions for the Navier-Stokes equations on the solid boundaries. Instead of that we define boundary layers, 5 cells adjacent to the top and bottom walls as the Boltzmann domain and the rest is the Navier-Stokes one, see Fig. 1 (b). After the domain decomposition the coupled solutions of the Boltzmann and Navier-Stokes equations match perfectly, see Fig. 2(a) for this small range of the Knudsen number.



(a) (b)

Fig. 1 (a) x component of velocity along the y axis at 2/3rd of the channel length for $Kn = 0.01101$ to 0.03303 from Boltzmann and Navier-Stokes solvers (b) A priori domain decomposition: 'red' or '+' = Navier-Stokes domain, 'green' or 'x' = Boltzmann domain

In the second test case, we increase the Knudsen number by changing different inlet and outlet pressures $168480Pa$ and $56160Pa$, respectively. This corresponds the Knudsen number varying 0.0408 to 0.12 from left to right boundaries. For this range of Knudsen numbers, we decrease the time step Δt to $2 \cdot 10^{-11} s$ for the Navier-Stokes solver. We are still in the slip regime and close to it, however, for this range of Knudsen numbers defining the boundary layers like in Fig. 1(b) does not provide the correct coupled solutions as shown in Fig. 2(b). Here, we observe that the coupled solution is close to the Navier-Stokes solution. In this case one may increase the size of boundary layers, but it is not clear how much one has to increase. So, we use the alternative strategy.



(a) (b)

Fig. 2 x component of velocity along the y axis at 2/3rd of the channel length. (a) for $Kn = 0.01101$ to 0.03303 (b) for $Kn = 0.0408$ to 0.12

The efficient way is to use a breakdown criterion to decompose the domains as suggested in [15] for steady problems. The idea is to solve first the Navier-Stokes equations everywhere until the steady state is reached. As we have seen in Fig. 2 (b), the Navier-Stokes solutions do not match with the Boltzmann solutions in this regime, however, they are somehow near to the Boltzmann ones. Then we apply the breakdown criterion $\|\phi\|$ suggested in [14] and decompose the domain. We assume, for example, if the value of $\|\phi\|$ at a cell is less than 0.01 the cell is defined as a Navier-Stokes cell, otherwise a Boltzmann one. In Fig. 3 the time evolution of the domain decompositions for the Knudsen numbers ranging from 0.01103 to 0.03303 at time different times are plotted. One can solve the Boltzmann and the Navier-Stokes equations in the corresponding domains. However, for the stationary solutions, it is sufficient to solve the Navier-Stokes equations until they reach the steady state and then to further use the domain decomposition and coupling method. After $t = 3 \cdot 10^{-8} s$ we reach the steady state of the Navier-Stokes equations and the domain decomposition does not change. After $t = 3 \cdot 10^{-8} s$ we solve both equations in their domains of validity until the final time. When we compare the figures Fig. 1(b) and Fig. 3 at time $t = 3 \cdot 10^{-8} s$, we see the Boltzmann domain is bigger in the latter figure. There is no unique values for this breakdown quantity. It depends upon the problem considered.

Now, for higher Knudsen numbers ranging from 0.0408 to 0.12 we observed that in the steady state the Navier-Stokes domain becomes smaller for the same criterion, see Fig. 4. Here the above coupling algorithm will not be the optimal one since we have a very small Navier-Stokes domain and we need additional effort to use the interface boundary conditions. Therefore, it is convenient to consider the entire domain as Boltzmann one with the initial conditions as stationary solutions of the Navier-Stokes equations. Then, we run for few more iterations and then start sampling the data.

The above results show that the coupling method may be relevant for regimes where the Knudsen number is less than 0.03.

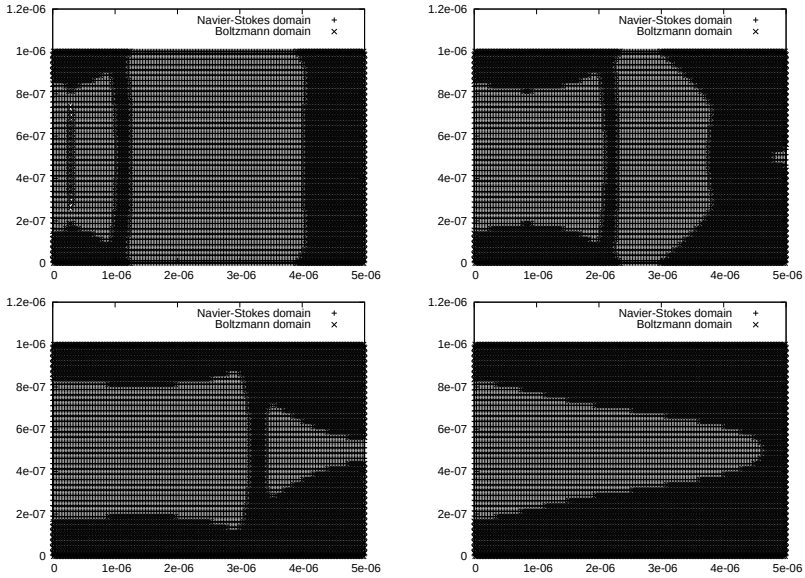


Fig. 3 Domain decomposition: 'red' or '+' = Navier-Stokes domain and 'green' or 'x' = Boltzmann domain after application of the breakdown criterion to solutions of the Navier-Stokes equations for the range $Kn = 0.01101$ to 0.03303 . Top rows are for $t = 3 \cdot 10^{-9} s$ and $t = 6 \cdot 10^{-9} s$ and the bottom rows are for $t = 9 \cdot 10^{-9} s$ and $t = 3 \cdot 10^{-8} s$.

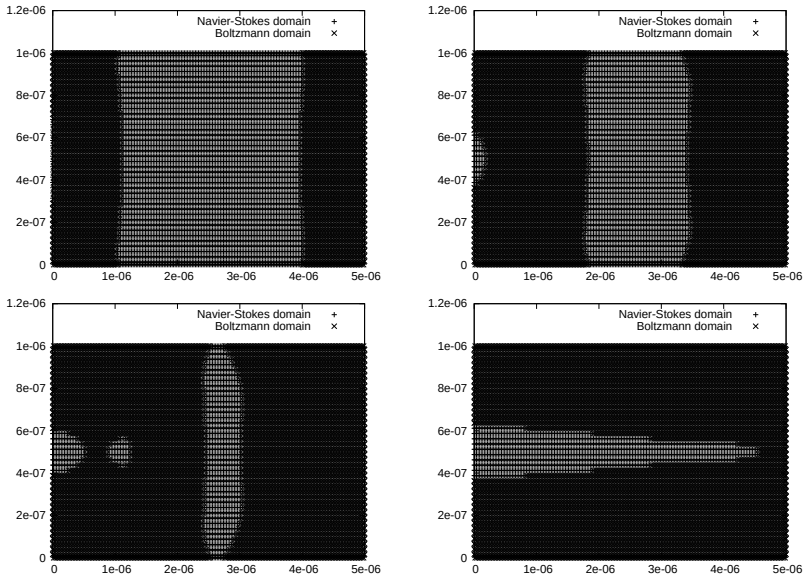


Fig. 4 Domain decomposition: 'red' or '+' = Navier-Stokes domain and 'green' or 'x' = Boltzmann domain after application of the breakdown criterion to stationary solutions of the Navier-Stokes equations for the range $Kn = 0.0408$ to 0.12. Top rows are for $t = 2 \cdot 10^{-9} s$ an $t = 4 \cdot 10^{-9} s$ and the bottom rows are for $t = 6 \cdot 10^{-9} s$ and $t = 2 \cdot 10^{-8} s$.

Acknowledgements This work was partially supported by the German Research Foundation (DFG), grant number KL1105/17-1.

References

1. Aktas, O., Aluru, N.R.: A combined continuum/dsmc technuque for multiscale analysis of microfluidic filters. J. Comput. Phys. **178**, 342–372 (2002)
2. Babovsky, H., Illner, R.: A convergence proof for nanbu's simulation method for the full boltzmann equation. SIAM J. Numer. Anal. **26**, 45–64 (1989)
3. Bardos, C., Golse, F., Levermore, D.: Fluid dynamic limits of kinetic equations. J. Stat. Phys. **63**, 323–344 (1991)
4. Bird, G.A.: Molecular Gas Dynamics and Direct Simulation of Gas Flows. Oxford University Press, New York (1994)
5. Caflish, R.: The fluid dynamical limit of the nonlinear boltzmann equation. Commun. Pure Appl. Math. **33**, 651–666 (1980)
6. Cercignani, C., Illner, R., Pulvirenti, M.: The Mathematical Theory of Dilute Gases. Springer-Verlag, Berlin (1994)
7. Chen S.and Weinan, E., Y., L., Shu, C.W.: A discontinuous galerkin implementation of a domain decomposition method for kinetic hydrodynamic coupling multiscale problems in gas dynamics and device simulations. J. Comput. Phys. **225**, 1314–1330 (2007)
8. Degond, P., Dimarco, G., Mieussens, L.: A moving interface method for dynamic kinetic-fluid coupling. J. Comput. Phys. **227**, 1176–1208 (2007)
9. Gad-el Hak, M.: The fluid mechanics of microdevices - the freeman scholar lecture. ASME J. Fluids Enggs. **121(403)**, 5–33 (1999)
10. Klar, A.: Domain decomposition for kinetic problems with nonequilibrium states. Eur. J. Mech., B/Fluids **15,2**, 203–216 (1996)
11. Le Tallec, P., Mallinger, F.: Coupling boltzmann and navier-stokes equations by half fluxes. J. Comput. Phys. **136**, 51–67 (1997)
12. Levermore, D., Morokoff, W.J., Nadiga, B.T.: Moment realizability and the validity of the navier-stokes equations for rarefied gas dynamics. Phys. Fluids **10(12)** (1998)
13. Neunzert, H., Struckmeier, J.: Particle methods for boltzmann equation. Acta Numer. **4**, 417–457 (1995)
14. Tiwari, S.: Coupling of the boltzmann and euler equations with automatic domain decomposition. J. Comput. Phys. **144**, 710–726 (1998)
15. Tiwari, S., Klar, A.: Coupling of the navier-stokes and the boltzmann equations with a meshfree particle and kinetic particle methods for a micro cavity. In: M. Griebel, M.A. Schweitzer (eds.) Meshfree Methods for Partial Differential Equations V, LNCSE 79, Berlin/Heidelberg, pp. 155–171. Springer (2011)
16. Tiwari, S., Klar, A., Hardt, S.: A particle-particle hybrid method for kinetic and continuum equations. J. Comput. Phys. **228**, 7109–7124 (2009)
17. Tiwari, S., Kuhnert, J.: Modeling of two phase flows with surface tension by finite pointset method (fpm). J. Comput. Appl. Math. **203**, 376–386 (2007)

Multiscale Finite Elements for Linear Elasticity: Oscillatory Boundary Conditions

Marco Buck[1], Oleg Iliev[1], and Heiko Andrä[1]

1 Introduction

Multiscale finite element methods (MsFEMs) have been widely used when solving elliptic PDEs with highly oscillating coefficients on multiple scales. Beyond their application in the upscaling framework [7, 8, 9, 3], they are often utilized for the construction of robust coarse spaces in the context of two-level overlapping domain decomposition preconditioners.

In [4, 2, 15] coarse basis functions are constructed by solving local generalized eigenvalue problems. The scalar multiscale finite element basis is used as a partition of unity to setup the spectral problems and allows the dimension of the resulting coarse space to be sufficiently low. The method guarantees robustness for various elliptic PDEs with respect to arbitrary coefficient variations. Another recent approach where generalized eigenvalue problems are solved in overlapping regions of local subdomains is presented in [13]. It provides applications to isotropic linear elasticity problems with robustness properties similar to them in [4, 2, 15].

For scalar elliptic PDEs it is shown in [5, 6] that oscillatory multiscale finite element coarse spaces ensure robustness for a large class of coefficient variations. This includes variations in the interior of coarse elements, but allows coefficient jumps also across coarse element boundaries when high contrast regions can be characterized as a union of disjoint islands.

A first application of the multiscale finite element method with (vector-valued) linear boundary conditions to linear elasticity (see also the adaptive method in [11]) is given in [1]. If material jumps occur only in the interior of coarse grid elements, uniform condition number bounds which do not depend on the contrast in the Young's modulus are obtained. However, the method fails to be robust when stiff inclusions touch coarse element bondaries. This motivates the construction of boundary conditions for the multiscale finite element basis which adapt to the heterogeneities in the PDE coefficients.

The outline of the paper is as follows. In Section 2 we state the equations of linear elasticity and briefly describe their discretization with vector-valued piecewise linear finite elements. The abstract two-level additive Schwarz method is summarized in Section 3. Section 4 contains the detailed introduction of the oscillatory multiscale finite element basis. Numerical results are presented in Section 5 and final conclusions are given in Section 6.

[1] Fraunhofer Institute for Industrial Mathematics (ITWM), Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany, e-mail: {Buck,Iliev,Andrae}@itwm.fraunhofer.de

2 Finite Element Discretization in Linear Elasticity

Let $\Omega \subset \mathbb{R}^d$ be a bounded, polyhedral ($d = 3$) or polygonal ($d = 2$) Lipschitz domain. The displacement field $u = (u_1, \ldots, u_d)^\top$ of a solid body in Ω, deformed under the action of a volume force f and a traction force t, is governed by the mixed BVP

$$\begin{aligned} -\operatorname{div}\sigma(u) &= f \ \text{ in } \Omega, \\ \sigma(u) &= C : \varepsilon(u) \ \text{ in } \Omega, \end{aligned} \tag{1}$$

where σ is the stress tensor, ε is the strain tensor and $C(x)$ is the fourth order elasticity tensor. The system in equation (1) is subject to the boundary conditions

$$u = 0 \text{ on } \Gamma_D, \ \ \sigma(u)n = t \text{ on } \Gamma_N,$$

where n is the unit outer normal vector on $\partial\Omega = \overline{\Gamma_D \cup \Gamma_N}$ with $\operatorname{meas}(\Gamma_D) > 0$.
Let $\mathscr{T}_h$ be a tetrahedral ($d = 3$) or triangular ($d = 2$) mesh and let $\Sigma_h(\bar{\Omega})$ denote the set of vertices in $\bar{\Omega}$. We introduce a finite element discretization u_h of displacements u on the space $\mathscr{V}^h := \operatorname{span}\{\varphi_k^{j,\mathrm{h}} : \bar{\Omega} \to \mathbb{R}^d, x^j \in \Sigma_h(\bar{\Omega}), k = 1, \ldots, d\}$ of continuous piecewise linear vector-valued functions on $\mathscr{T}_h$. Assuming enough regularity, the discretization leads to a symmetric positive definite linear system $A\mathbf{u} = \mathbf{f}$ (see e.g. [10] for more details).

3 Overlapping Domain Decomposition Preconditioners

We are interested in constructing two-level overlapping domain decomposition preconditioners for the linear system which are robust w.r.t. mesh parameters and variations in the PDE coefficients. They combine local solves on overlapping subdomains $\{\Omega_i, i = 1, \ldots, N\}$ (with overlap-width $\delta > 0$) and a global solve on a coarse grid $\mathscr{T}_H$. Let $\mathscr{V}^0 \subset \mathscr{V}_0^h$ be a coarse space defined on $\mathscr{T}_H$ and let $\mathscr{V}^i = \mathscr{V}^h(\Omega_i)$ be the space of vector-valued linear basis functions on $\mathscr{T}_h$ which are supported in Ω_i, $i = 1, \ldots, N$. The action of the two-level additive Schwarz preconditioner is defined implicitly by

$$M_{\mathrm{AS}}^{-1} = R_0^\top A_0^{-1} R_0 + \sum_{i=1}^{N} R_i^\top A_i^{-1} R_i,$$

where R_i, $i = 0, \ldots, N$ is the restriction operator from $\mathscr{V}^h$ to $\mathscr{V}^i$ and $A_i = R_i A R_i^\top$ is the corresponding submatrix of A (cf. [14]). We assume here that $\mathscr{T}_H$ also consists of tetrahedra ($d = 3$) or triangles ($d = 2$), each of which consists of a union of fine elements $\tau \in \mathscr{T}_h$. For any $D \subset \bar{\Omega}$, we denote by $\Sigma_H(D)$ the set of nodes of $\mathscr{T}_H$ in D and $\mathscr{N}_H(D)$ is the corresponding index-set of coarse nodes.

4 Multiscale Finite Elements for Linear Elasticity

Multiscale basis functions with oscillatory boundary conditions are introduced for scalar elliptic PDEs in [7] to reflect the heterogeneities in the PDE coefficients also across coarse element boundaries. In this section we present the extension to linear elasticity. We define the multiscale basis and introduce suitable coordinate transformations that allow the derivation of the equations which govern the boundary data of the oscillatory multiscale basis on general meshes. On composites with isotropic constituents, we present the construction in detail. We denote by $\bar{\omega}_p := \{T \subseteq \mathscr{T}_H : p \in \mathscr{N}_H(T)\}$ the union of coarse elements which share the node $x^p \in \Sigma_H(\bar{\Omega})$. For any $p \in \mathscr{N}_H(\bar{\Omega})$ and $m \in \{1,\dots,d\}$, the oscillatory multiscale basis function $\mathscr{V}^h \ni \phi_m^{p,\mathrm{MsO}} : \omega_p \to \mathbb{R}^d$, is defined such that for $T \subset \bar{\omega}_p$,

$$\begin{aligned} \operatorname{div}(\mathrm{C} : \varepsilon(\phi_m^{p,\mathrm{MsO}})) &= 0 \qquad \text{in } T, \\ \phi_m^{p,\mathrm{MsO}} &= \eta_m^{p,T} \quad \text{on } \partial T, \end{aligned} \tag{2}$$

where the oscillatory boundary data $\eta_m^{p,T} : \partial T \to \mathbb{R}^d$ are continuous and compatible, i.e. $\eta_m^{p,T} = \eta_m^{p,T'}$ on $\partial T \cap \partial T' \subset \bar{\Omega}$ for $T, T' \in \mathscr{T}_H$. We impose the vector-valued nodal constraints

$$\eta_{mk}^{p,T}(x^q) = \delta_{pq}\delta_{mk},\ x^q \in \mathscr{N}_H(T),\ k \in \{1,\dots,d\} \tag{3}$$

and show how $\eta_m^{p,T} = (\eta_{m1}^{p,T},\dots,\eta_{md}^{p,T})^\top$ is derived in Section 4.2 and 4.3.

4.1 Coordinate Transformation

The boundary data $\eta_m^{p,T}$ in equation (2) are extracted by solving a restricted version of the PDE (1) to the coarse element boundary which implies that $\phi_m^{p,\mathrm{MsO}}|_{\partial T}$ is independent of the coordinate in the direction normal to ∂T. To make the construction applicable to edges and faces of $T \in \mathscr{T}_H$ which are not aligned with or perpendicular to one of the coordinate axis, we apply a suitable coordinate transformation of the Cartesian coordinate system with basis $\{e^1,\dots,e^d\}$ to a (right handed) coordinate system with orthonormal basis $\{\hat{e}^1,\dots,\hat{e}^d\}$. W.l.o.g., for any

edge $\mathscr{E}$: we introduce the rotated coordinate system such that $\hat{e}^1$ is parallel to $\mathscr{E}$
face $\mathscr{F}$: we introduce the rotated coordinate system such that the normal vector n on $\mathscr{F}$ is parallel to one of the coordinate axis, i.e. $\hat{e}^3 = n$.

Let $\hat{x}_1,\dots,\hat{x}_d$ be the coordinates of $x = (x_1,\dots,x_d)^\top$ w.r.t. the transformed basis. The coordinate transformation can be described by a linear map $\Theta : T \to \mathbb{R}^d$, $\hat{x} = \Theta x$ with $\theta_{ij} = \hat{e}^i \cdot e^j$, $1 \le i, j \le d$. The elasticity coefficients of the stiffness tensor $\hat{C}$ transform under the rotation of the coordinate system to $\hat{c}_{ijkl} = \sum_{p,q,r,s=1}^{d} \theta_{ip}\,\theta_{jq}\,\theta_{kr}\,\theta_{ls}\,c_{pqrs}$ (cf. [12]).

4.2 Equations Governing the Oscillatory Boundary Data

Using the rotated coordinate system in Section 4.1, we derive the reduced problems on a face $\mathscr{F}$ of $T \in \mathscr{T}_H$ for the system of anisotropic linear elasticity. The components of the elasticity operator in equation (1) read

$$\sum_{j=1}^{d} \partial_j \sigma_{ij}(u) = \sum_{j=1}^{d} \partial_j \Big(\sum_{k,l=1}^{d} c_{ijkl}\, \varepsilon_{kl}(u) \Big). \tag{4}$$

Forcing that $\hat{\phi}_m^{p,\mathrm{MsO}} = \hat{\eta}_m^{p,T}(\hat{x}_1, \ldots, \hat{x}_{d-1})$ is independent of $\hat{x}_d$ on $\mathscr{F}$ and using the symmetry $\hat{c}_{ijkl} = \hat{c}_{ijlk}$ of the stiffness tensor, we obtain by using $\hat{\varepsilon}_{kl}(\hat{u}) = \frac{1}{2}(\hat{\partial}_k \hat{u}_l + \hat{\partial}_l \hat{u}_k)$ in the rotated coordinate system

$$\begin{aligned}
\sum_{j=1}^{d} \hat{\partial}_j \hat{\sigma}_{ij}(\hat{\eta}_m^{p,T}) &= \sum_{j=1}^{d-1} \hat{\partial}_j \Big(\sum_{k,l=1}^{d} \hat{c}_{ijkl}\, \hat{\varepsilon}_{kl}(\hat{\eta}_m^{p,T}) \Big) \\
&= \sum_{j=1}^{d-1} \hat{\partial}_j \Big(\sum_{k,l=1}^{d-1} \hat{c}_{ijkl}\, \hat{\varepsilon}_{kl}(\hat{\eta}_m^{p,T}) + 2 \sum_{k=1}^{d-1} \hat{c}_{ijkd}\, \hat{\varepsilon}_{kd}(\hat{\eta}_m^{p,T}) \Big) \\
&= \sum_{j=1}^{d-1} \hat{\partial}_j \Big(\sum_{k,l=1}^{d-1} \hat{c}_{ijkl}\, \hat{\varepsilon}_{kl}(\hat{\eta}_m^{p,T}) \Big) \qquad (5) \\
&\quad + \sum_{j=1}^{d-1} \hat{\partial}_j \Big(\sum_{k=1}^{d-1} \hat{c}_{ijkd}\, \hat{\partial}_k \hat{\eta}_{md}^{p,T} \Big). \qquad (6)
\end{aligned}$$

While equation (5) affects exclusively the first two components of $\hat{\eta}_m^{p,T}$, equation (6) acts only on the third component of the oscillatory boundary data on $\mathscr{F}$. For an anisotropic stiffness tensor, a reduced system needs to be solved on $\mathscr{F}$ in which the three components of $\hat{\eta}_{m2}^{p,T}$ are coupled. Having a deeper look at the entries of the stiffness tensor, the systems in (5) and (6) are fully decoupled for an orthotropic material whose symmetry axes are normal to $\hat{e}^1, \ldots, \hat{e}^d$. Particularly, the components $\hat{\eta}_{m1}^{p,T}$ and $\hat{\eta}_{m2}^{p,T}$ on $\mathscr{F}$ are then governed by a 2D system of linear elasticity (see (5)), while the component $\hat{\eta}_{md}^{p,T}$ normal to $\mathscr{F}$ is governed by a scalar second order elliptic PDE (see (6)). Analogously, on an edge $\mathscr{E}$, we can deduce that the boundary data $\hat{\eta}_m^{p,T}(\hat{x}_1)$ are governed by scalar second order PDEs in each particular component which may, again, be coupled in the anisotopic case.

4.3 Oscillatory Boundary Conditions for Isotropic Linear Elasticity

Given the formulation of the reduced problems in a suitable coordinate system, we summarize the procedure of computing boundary data $\eta_m^{p,T}$ on the faces and edges of T, assuming that the stiffness tensor is isotropic. Its components are given by

$c_{ijkl} = \lambda\,\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$, where $\mu > 0$ and $\lambda \geq -\frac{2}{3}\mu$ are the Lamé coefficients of the material (see e.g. [10]) which we assume here to be piecewise constant in $\tau \in \mathscr{T}_h$. Note that the material coefficients are not uniquely determined on ∂T, a proper averaging (e.g. by taking their maximum values) in the adjacent elements $\tau \in \mathscr{T}_h$ is required.
From (5) and (6), together with $\hat{\eta}_m^{p,T} = \hat{\eta}_m^{p,T}(\hat{x}_1)$ along the edge $\mathscr{E}$, the reduced problem in rotated coordinates reads

$$\begin{aligned} \hat{\partial}_1\Big((\lambda+2\mu)\,\hat{\partial}_1\hat{\eta}_{m1}^{p,T}\Big) &= 0 \text{ on } \mathscr{E}, \\ \hat{\partial}_1\Big(\mu\,\hat{\partial}_1\hat{\eta}_{mk}^{p,T}\Big) &= 0 \text{ on } \mathscr{E},\ k = 2,3. \end{aligned} \tag{7}$$

It needs to be equipped with the boundary conditions defined in (3). Let us assume that $\mathscr{E} = \mathscr{E}_{p_1p_2}$ connects the two nodes $x^{p_1} = x^p, x^{p_2} \in \Sigma_H(\bar{\Omega})$, then we impose

$$\begin{aligned} \hat{\eta}_m^{p,T}(\hat{x}^{p_1}) &= \Theta e^m, \\ \hat{\eta}_m^{p,T}(\hat{x}^{p_2}) &= (0,0,0)^\top. \end{aligned} \tag{8}$$

In order to grasp immediately that the boundary data on a face $\mathscr{F}$ are governed by a reduced elasticity system in the first two components and a scalar elliptic problem in the component normal to $\mathscr{F}$, we state the equations governing the reduced problem under the assumption that λ and μ are piecewise constant on $\mathscr{F}$. This allows to simplify the notation of the reduced system without affecting its weak formulation. According to equation (5) and (6), the reduced system reads

$$\begin{aligned} \mu\,(\hat{\partial}_{11}\hat{\eta}_{m1}^{p,T} + \hat{\partial}_{22}\hat{\eta}_{m1}^{p,T}) + (\lambda+\mu)(\hat{\partial}_{11}\hat{\eta}_{m1}^{p,T} + \hat{\partial}_{12}\hat{\eta}_{m2}^{p,T}) &= 0 \text{ a.e. on } \mathscr{F}, \\ \mu\,(\hat{\partial}_{11}\hat{\eta}_{m2}^{p,T} + \hat{\partial}_{22}\hat{\eta}_{m2}^{p,T}) + (\lambda+\mu)(\hat{\partial}_{21}\hat{\eta}_{m1}^{p,T} + \hat{\partial}_{22}\hat{\eta}_{m2}^{p,T}) &= 0 \text{ a.e. on } \mathscr{F}, \\ \mu\,(\hat{\partial}_{11}\hat{\eta}_{m3}^{p,T} + \hat{\partial}_{22}\hat{\eta}_{m3}^{p,T}) &= 0 \text{ a.e. on } \mathscr{F}. \end{aligned} \tag{9}$$

Let $\mathscr{F} = \mathscr{F}_{p_1p_2p_3}$ contain the coarse nodes x^{p_1}, x^{p_2} and x^{p_3}. Then the three edges $\mathscr{E}_{p_1p_2}$, $\mathscr{E}_{p_1p_3}$ and $\mathscr{E}_{p_2p_3}$ form the 2D boundary of the face $\mathscr{F}$. The system in (9) is subject to the boundary conditions

$$\hat{\eta}_m^{p,\mathscr{F}}\,|_{\mathscr{E}_{p_kp_l}} = \hat{\eta}_m^{p,\mathscr{E}_{p_kp_l}} \quad 1 \leq k < l \leq 3,$$

where $\hat{\eta}_m^{\mathscr{E}_{p_kp_l}}$ is the solution of the BVP in (7) and (8) on the edge $\mathscr{E}_{p_kp_l}$ in the coordinate system w.r.t. $\mathscr{F}$ and $\hat{\eta}_m^{p,\mathscr{D}}$ denotes the restriction of $\hat{\eta}_m^{p,T}$ to $\mathscr{D} \subset \partial T$. Note that the rotated coordinate systems differ for any face and edge. Once the boundary data are computed on and edge or a face, they should be transformed to the original coordinate system.

4.4 Properties of the Oscillatory Multiscale Basis

As shown in [1], the multiscale basis with vector-valued linear boundary data (*MsL*) recovers all rigid body modes. If no material jumps occur on the boundaries of coarse elements, it can be shown that $\phi_m^{p,\mathrm{MsO}} = \phi_m^{p,\mathrm{MsL}}$. Prescribing homogeneous material parameters, both multiscale bases coincide with the vector-valued linear coarse basis. Furthermore, the construction of the oscillatory multiscale basis guarantees that the rigid body translations are contained in the coarse space. In general, not all the rigid body rotations are preserved exactly on the coarse element boundaries. The complexity of computing $\phi_m^{p,\mathrm{MsO}}$ is of the same asymptotic order $O\big(d(\frac{H}{h})^d\big)$ as for $\phi_m^{p,\mathrm{MsL}}$, with a small additional cost that is one order of $\frac{H}{h}$ cheaper.

5 Numerical Results

In this section we present numerical examples on a binary composite. We apply different coarsening strategies for the two-level additive Schwarz preconditioner, including a vector-valued linear coarse space as well as multiscale coarse spaces with linear and oscillatory boundary conditions. We perform the simulations on a domain $\overline{\Omega} = [0,1] \times [0,1] \times [0,L], L > 0$, using regular fine and coarse triangular meshes $\mathscr{T}_h$ and $\mathscr{T}_H$ of equal structure with uniform mesh size h and H, respectively. Both meshes are constructed from an initial voxel geometry by decomposing each voxel into five tetrahedra. In the experiments we show condition numbers as well as iteration numbers of the PCG algorithm. The stopping criterion is set to reduce the preconditioned initial residual by 6 orders of magnitude.

The medium consists of an isotropic matrix material with coefficients ($\mu_{\mathrm{mat}} = 1$, $\lambda_{\mathrm{mat}} = 1$) and contains inclusions ($\mu_{\mathrm{inc}}, \lambda_{\mathrm{inc}}$) which are positioned equally in each coarse block of size $H \times H \times H$ as shown in Fig. 1. The distribution of the inclusions as well as the boundaries of the coarse tetrahedra are shown in more detail in Fig. 2. At each slice in the plane normal to X_1 and X_2 the position of the inclusions above and below this level are indicated in dark and shaded red, respectively. Each inclusion touches or crosses coarse element boundaries while one inclusion in the center is isolated in the interior of a coarse element. Table 1 shows the condition

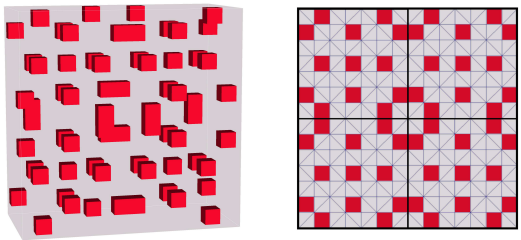


Fig. 1 Binary composite; matrix material (grey) and inclusions (red); discretization in 14x14x7 voxels (left); 2D-projection onto the (X_1, X_2)-plane with position of the inclusion (right); each coarse block is decomposed in five tetrahedra;

and iteration numbers for the three coarsening strategies under the variation of the material contrast $\Delta_E := \mu_{\text{inc}}/\mu_{\text{mat}} = \lambda_{\text{inc}}/\lambda_{\text{mat}}$. For $\Delta_E > 1$, condition and iteration numbers for vector-valued linear and multiscale coarse space with linear boundary conditions grow with the contrast in the material coefficients, where the latter does not perform noticeably better than the linear coarse space. The multiscale coarse basis functions with oscillatory boundary conditions are bounded in energy and show coefficient-independent bounds of the condition number. For $\Delta_E < 1$, each coarse space performs well.

Table 1 Condition numbers κ and iteration numbers (#it) of precond. matrix for $H = 7h$, $\delta = 2h$

Δ_E	Lin	MsL	MsO
10^{-9}	26 (28)	26 (28)	26 (28)
10^{-6}	26 (28)	26 (28)	26 (28)
10^{-3}	26 (28)	26 (28)	26 (28)
10^{0}	25 (27)	25 (27)	25 (27)
10^{3}	426 (91)	233 (76)	25 (27)
10^{6}	965 (102)	955 (104)	25 (27)
10^{9}	970 (102)	955 (104)	25 (27)

6 Conclusions

In this study, we extended the oscillatory multiscale finite element method as introduced in [7] to the PDE system of anisotropic linear elasticity. We derived the reduced system which governs the oscillatory boundary data in a general setting which allows their construction on triangular, tetrahedral, quadrilateral and hexahedral coarse meshes. We applied the coarse basis in the context of two-level additive Schwarz domain decomposition preconditioners. Numerical results are presented on a tetrahedral mesh for isotropic composites where inclusions touch the coarse

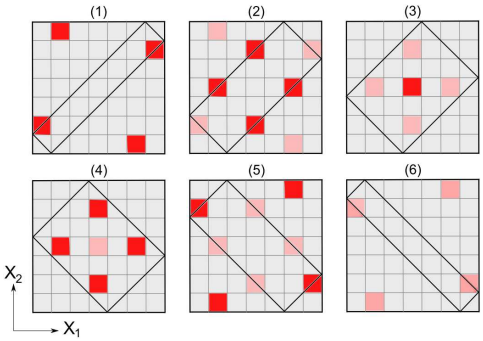


Fig. 2 2D-slices (at $X_3 = l\,h$, $l \in \{1,\ldots,6\}$) of a coarse block of $7\times7\times7$ voxels of the medium in Fig. 1 ; boundaries of coarse tetrahedral elements (black), matrix material (grey) and 1x1x1 inclusions (red); inclusions touch the slice from below (shaded red) or top (dark red); inclusions touch coarse element boundaries

element boundaries. We observed condition number bounds of the preconditioned linear system which are independent of the contrast in the Young's modulus in the inclusions.

It is easy to verify (see e.g. [1]) that the computation of a multiscale finite element basis is more costly on quadrilateral and hexahedral coarse meshes than on their triangular and tetrahedral counterparts (by a factor of $\frac{4}{3}$ in 2D and a factor of 2 in 3D). However, we may point out that, especially for applications in three spatial dimensions, using hexahedral coarse meshes may be beneficial for the robustness of the overall method as it reduces the amount of element boundaries which are introduced when tetrahedral coarse meshes are used.

Acknowledgements The authors would like to thank Prof. Yalchin Efendiev and Prof. Victor Calo for fruitful discussions and their valuable comments on the subject of this manuscript.

References

1. Buck, M., Iliev, O., Andrä, H.: Multiscale finite element coarse spaces for the application to linear elasticity. Cent. Eur. J. Math. **11**(4), 680–701 (2013)
2. Efendiev, Y., Galvis, J., Lazarov, R., Willems, J.: Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms. Math. Model. Numer. Anal. **46**, 1175–1199 (2012)
3. Efendiev, Y., Hou, T.: Multiscale finite element methods, theory and applications, 4 edn. (2009)
4. Galvis, J., Efendiev, Y.: Domain decomposition preconditioners for multiscale flows in high-contrast media: reduced dimension coarse spaces. Multiscale Model. Simul. **8**(5), 1621–1644 (2010)
5. Graham, I.G., Lechner, P.O., Scheichl, R.: Domain decomposition for multiscale PDEs. Numer. Math. **106**, 589–626 (2007)
6. Graham, I.G., Scheichl, R.: Robust domain decomposition algorithms for multiscale PDEs. Numer. Methods Partial Differ. Equ. **23**, 859–878 (2007)
7. Hou, T.Y., Wu, X.H.: A multiscale finite element method for elliptic problems in composite materials and porous media. J. Comput. Phys. **134**, 169–189 (1997)
8. Hou, T.Y., Wu, X.H.: A multiscale finite element method for PDEs with oscillatory coefficients. Note Num. Fl. **70**, 58–69 (1999)
9. Hou, T.Y., Wu, X.H., Cai, Z.: Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. Math. Comp. **68**, 913–943 (1999)
10. Hughes, T.J.R.: The finite element method: Linear static and dynamic finite element analysis. Prentice-Hall, Inc. (1987)
11. Millward, R.: A new adaptive multiscale finite element method with applications to high contrast interface problems. Ph.D. thesis, University of Bath (2011)
12. Norris, A.: Euler-rodrigues and Cayley formulas for rotation of elasticity tensors. Math. Mech. Solids **13**, 465–498 (2008)
13. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R.: Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. Tech. Rep. 2011-07, University of Linz, Institute of Computational Mathematics (2011)
14. Toselli, A., Widlund, O.: Domain decomposition methods, algorithms and theory. Springer (2005)
15. Willems, J.: Robust multilevel methods for general symmetric positive definite operators. Tech. Rep. 2012-06, RICAM Institute for Computational and Applied Mathematics (2012)

Inexact BDDC methods for the cardiac Bidomain model

Stefano Zampini[1]

1 Introduction

The cardiac Bidomain model consists in a reaction-diffusion system of PDEs for the intra- and extra-cellular cardiac potentials coupled with a nonlinear system of ODEs accounting for the cellular model of ionic currents. Fully implicit methods in time have been considered in a few studies, see e.g. [16] and references therein. As in most of previous work (see [18] for a review), in this study we consider an Implicit-Explicit operator splitting technique in order to separate the part of the system of PDEs describing diffusion of cardiac potentials from the large and stiff nonlinear system of ODEs accounting for the reaction terms. The resulting space-time discretization of the so-called parabolic-parabolic Bidomain operator leads to a large, sparse, symmetric positive semidefinite linear system which must be solved at each time step of a cardiac beat simulation using a Krylov subspace method. Given a component by component finite element discretization of the cardiac potentials, the coefficient matrix of the linear system to be solved is

$$\widehat{K} = \begin{bmatrix} A_i & 0 \\ 0 & A_e \end{bmatrix} + \frac{\chi}{\delta_t} \begin{bmatrix} M & -M \\ -M & M \end{bmatrix} \tag{1}$$

where δ_t is the value of the time step and χ the membrane capacitance per unit volume; M and $A_{i,e}$ are the mass and stiffness matrices with entries

$$\{M\}_{rs} = \int_\Omega \phi_h^r \phi_h^s, \quad \{A_{i,e}\}_{rs} = \int_\Omega D_{i,e} \nabla \phi_h^r \cdot \nabla \phi_h^s,$$

where for sake of simplicity the same finite element basis $\{\phi_h^j\}$ is considered for each cardiac potential. Anisotropic conductivity tensors $D_i(x)$ and $D_e(x)$ model propagation of electrical signals with orthotropic anisotropy

$$D_{i,e}(x) = \sum_{j=1}^{3} \sigma_j^{i,e}(x) \mathbf{a}_j(x) \mathbf{a}_j(x)^T,$$

with $\sigma_j^{i,e}(x) > 0$ the conductivity coefficient of the intra- and extra-cellular media measured along the orthonormal triplet $\{\mathbf{a}_j(x)\}_{j=1}^3$ describing cardiac fiber rotation [9]. For additional details on the operator splitting technique adopted and the diffusion tensors, see [6].

[1] CINECA, SuperComputing Applications and Innovations dept., Rome branch, Via dei Tizii 6, 00185 Rome (Italy) s.zampini@cineca.it

Many different preconditioners have been already proposed for the efficient iterative solution of the Bidomain model in its parabolic-parabolic formulation (1). Among them, we mention block Jacobi preconditioners [6], algebraic multigrid [13, 14], multilevel Schwarz preconditioners [11, 15, 12] and balancing Neumann-Neumann methods [19]. An exact BDDC algorithm and a FETI-DP method have been constructed, analyzed and experimentally validated by the Author in [20].

2 Inexact BDDC preconditioner

Following the framework of substructuring algorithms [17], the cardiac domain Ω is decomposed into N non-overlapping open Lipschitz subdomains Ω_j of diameter H_j, forming a coarse conforming finite element partition of Ω and naturally defining the interface, i.e.

$$\overline{\Omega} = \bigcup_{i=j}^{N} \overline{\Omega}_j, \quad \Gamma = \bigcup_{j \neq k} \partial\Omega_j \cap \partial\Omega_k, \quad \Gamma_j = \partial\Omega_j \cap \Gamma.$$

A triangulation is introduced in each subdomain with matching finite element nodes on the boundaries of adjacent subdomains across the interface. As usual in non-overlapping literature, the finite elements space defined on Ω_j will be denoted by $\mathbf{W}^{(j)}$ and it is further split into its interior (labeled by I) and interface (Γ) parts; the following spaces should then be introduced

$$\mathbf{W}^{(j)} = \mathbf{W}_I^{(j)} \bigoplus \mathbf{W}_\Gamma^{(j)}, \quad \mathbf{W} = \prod_{j=1}^{N} \mathbf{W}^{(j)}, \quad \mathbf{W}_I = \prod_{j=1}^{N} \mathbf{W}_I^{(j)},$$

together with the subspace $\widehat{\mathbf{W}} \subset \mathbf{W}$ of continuous functions. Within the non-overlapping framework, a global matrix is never assembled explicitly; instead a Bidomain linear matrix $K^{(j)}$ is assembled on each subdomain and reordered as

$$\begin{bmatrix} K_{II}^{(j)} & K_{I\Gamma}^{(j)} \\ K_{I\Gamma}^{(j)^T} & K_{\Gamma\Gamma}^{(j)} \end{bmatrix}.$$

The unassembled global matrix defined on $\mathbf{W}$ can thus be defined as $K = \mathrm{diag}(K^{(j)})$; similarly, $K_{II} = \mathrm{diag}(K_{II}^{(j)})$.

The exact BDDC preconditioner for matrix $\widehat{K}$ can be formulated as (see [8, 10])

$$M_{BDDC}^{-1} = M_I^{-1} + (I - M_I^{-1}\widehat{K}) M_\Gamma^{-1} (I - \widehat{K} M_I^{-1}),$$

where

$$M_I^{-1} = R_I^T K_{II}^{-1} R_I, \quad M_\Gamma^{-1} = R_D^T (P_{coarse} + P_{local}) R_D,$$

with R_I the restriction operator from $\widehat{\mathbf{W}}$ to $\mathbf{W}$ and R_D the scaled restriction operator from $\widehat{\mathbf{W}}$ to $\mathbf{W}$ built using a suitable partition of unity [20]. The coarse term of the preconditioner can be defined by

$$P_{coarse} = \Psi K_c^{-1} \Psi^T, \quad K_c = \Psi^T K \Psi,$$

with the coarse primal basis function matrix given by the solution of the following minimization problem posed on $\mathbf{W}$

$$\Psi = \arg\min w^T K w, \text{ s.t. } Cw = I,$$

where I is the identity matrix and C is the block diagonal matrix of BDDC constraints which ensures the continuity of coarse basis functions at primal degrees of freedom. The action of the local term of the preconditioner is given by

$$\begin{bmatrix} K & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} P_{local}\, g \\ \mu \end{bmatrix} = \begin{bmatrix} g \\ 0 \end{bmatrix}.$$

The application of the BDDC preconditioner requires the solution of the block diagonal Dirichlet and Neumann problems given by the matrices K_{II} and K_{rr} respectively, where K_{rr} is obtained from K by removing the matrix entries related to the subdomain vertices belonging to the coarse primal space [8].

It is well known that the local problems defined by the BDDC preconditioner can be bottlenecks in three dimensions, since direct factorizations require too much time and memory if the number of degrees of freedom in any subdomain is large; also, backward and forward substitution algorithms do not map well on modern architectures and accelerators. A possible solution consists in using multigrid preconditioners as black-box inexact solvers for the local Dirichlet and Neumann problems as proposed by Dohrmann [8]; the approach preserves scalability and quasi-optimality of the exact BDDC method provided a sufficient quality of the inexact solvers.

An approximate BDDC preconditioner can be constructed as follows: let $\widehat{K}^\flat$ be the matrix which is equal to $\widehat{K}$ except for the coupling of the interior degrees of freedom and let $K^\sharp$ be the matrix equal to K except for the blocks related to the Neumann problem of the BDDC preconditioner, i.e.

$$\widehat{K}^\flat = \begin{bmatrix} K_{II}^\flat & K_{I\Gamma} \\ K_{I\Gamma}^T & K_{\Gamma\Gamma} \end{bmatrix}, \quad K^\sharp = \begin{bmatrix} K_{rr}^\sharp & K_{rv} \\ K_{rv}^T & K_{vv} \end{bmatrix}.$$

In practice, matrices $K_{II}^\flat$ and $K_{rr}^\sharp$ are not explicitly known, since they represent an approximation of the exact matrices through the multigrid process.

Inexact solvers can be obtained in such a way that $K^\flat$ and $K^\sharp$ will be spectrally equivalent to the exact matrices

$$\gamma_1 g^T \widehat{K} g \le g^T \widehat{K}^\flat g \le \gamma_2 g^T \widehat{K} g \quad \forall g \in \widehat{\mathbf{W}}, \tag{2}$$

$$\alpha_1 g^T K g \le g^T K^\sharp g \le \alpha_2 g^T K g \quad \forall g \in \mathbf{W}. \tag{3}$$

where $0 < \gamma_1 \leq \gamma_2$ and $0 < \alpha_1 \leq \alpha_2$ are constants independent on h and $H = \max_j H_j$. A priori estimates for the latter constants are not required for the implementation, but they can be estimated by conjugate gradient iterations. In addition, if the matrix $\widehat{K}$ is singular as for the Bidomain model, matrices $K^\flat$ and $K^\sharp$ should satisfy the so called null space property

$$\ker(\widehat{K}^\flat) = \ker(\widehat{K}),\ \ker(K^\sharp) = \ker(K).$$

Given a candidate preconditioner P_{II}^{-1} for $K_{II}^{\flat\,-1}$, the following correction was proposed in [8] to satisfy the null space property

$$K_{II}^{\flat\,-1} = N_I (N_I^T K_{II} N_I)^{-1} N_I^T + E_I^T P_{II}^{-1} E_I, \tag{4}$$

where

$$E_I = I - K_{II} N_I (N_I^T K_{II} N_I)^{-1} N_I^T,$$

with I the identity matrix and N_I the restriction of $\ker(\widehat{K})$ to the interior degrees of freedom. The same argument holds true for the Neumann problem, thus

$$K_{rr}^{\sharp\,-1} = N_r (N_r^T K_{rr} N_r)^{-1} N_r^T + E_r^T P_{rr}^{-1} E_r, \tag{5}$$

where

$$E_r = I - K_{rr} N_r (N_r^T K_{rr} N_r)^{-1} N_r^T,$$

with P_{rr}^{-1} a candidate preconditioner for $K_{rr}^{\sharp\,-1}$.

The action of the approximate BDDC preconditioner can then be defined as

$$\widetilde{M}_{BDDC}^{-1} = M_I^{\flat\,-1} + (I - M_I^{\flat\,-1} \widehat{K}^\flat) M_\Gamma^{\sharp\,-1} (I - \widehat{K}^\flat M_I^{\flat\,-1}),$$

where the superscript $\flat$ (respectively $\sharp$) denote quantities obtained by replacing the matrix $\widehat{K}$ (resp. K) by $K^\flat$ (resp. $K^\sharp$) in the construction of the BDDC operator. In other words,

$$M_I^{\flat\,-1} = R_I^T K_{II}^{\flat\,-1} R_I, \quad M_\Gamma^{\sharp\,-1} = R_D^T \left[P_{coarse}^\sharp + P_{local}^\sharp \right] R_D,$$

with

$$P_{coarse}^\sharp = \Psi^\sharp K_c^{\sharp\,-1} \Psi^{\sharp\,T}, \quad K_c^\sharp = \Psi^{\sharp\,T} K^\sharp \Psi^\sharp,$$

and the block saddle point matrix is modified as

$$\begin{bmatrix} K^\sharp & C^T \\ C & 0 \end{bmatrix}.$$

For further details on the inexact approach considered, see [8].

The following theorem holds (see [8] for the proof).

Theorem 1. *The condition number of the approximate BDDC preconditioner can be bounded from above by the condition number of the exact BDDC preconditioner as*

$$\kappa_2(\widetilde{M}^{-1}_{BDDC}\widehat{K}) \leq C\frac{\alpha_2\gamma_2^3}{\alpha_1\gamma_1^3}\kappa_2(M^{-1}_{BDDC}\widehat{K}),$$

where γ_1 and γ_2 are given by (2), α_1 and α_2 by (3) and C is a constant independent of the parameters of the spatial discretization h and H and the number of subdomains N. Moreover, if the coarse problem $A_c^{\sharp}$ is solved inexactly by the action of a preconditioner $A_c^{\sharp\sharp^{-1}}$ satisfying

$$\beta_1 g^T A_c^{\sharp^{-1}} g \leq g^T A_c^{\sharp\sharp^{-1}} g \leq \beta_2 g^T A_c^{\sharp^{-1}} g,$$

with $0 < \beta_1 \leq \beta_2$, it will hold

$$\kappa_2(\widetilde{M}^{-1}_{BDDC}\widehat{K}) \leq C\frac{\max\{1,\beta_2\}\alpha_2\gamma_2^3}{\min\{1,\beta_1\}\alpha_1\gamma_1^3}\kappa_2(M^{-1}_{BDDC}\widehat{K}).$$

A quasi-optimal bound for the condition number of the exact BDDC method for the Bidomain model in the parabolic-parabolic form has been proved in [20].

Theorem 2. *Let the BDDC coarse primal space be spanned by the vertex nodal finite element functions and the edge cut-off functions. Then, for the three-dimensional Bidomain model, it will hold*

$$\kappa_2(M^{-1}_{BDDC}\widehat{K}) \leq C(1+\log(H/h))^2,$$

with $H = \max_j H_j$ and C a constant independent of h, H, N and possible jumps in conductivity coefficients $\sigma_k^{(i,e)}$ of the Bidomain operator aligned with Γ.

3 Numerical resuts

In this Section parallel numerical experiments are presented for a parallellepipedal domain Ω subdivided into $N = N_x \times N_y \times N_z$ subdomains. Each Ω_j is discretized by low-order Q^1 finite elements, i.e. conforming hexaedral shape-regular isoparametric tri-linear finite elements of characteristic diameter h. The linear system (1) is solved by the preconditioned conjugate gradient (PCG) algorithm with a zero initial guess and stopping criterion $\|r_k\|_2/\|r_0\|_2 \leq 10^{-6}$, where r_k is the preconditioned residual at the kth iterate. The right-hand side is always random and uniformly distributed. Extreme eigenvalues of the preconditioned operators, denoted by λ_m and λ_M in the following, are estimated using the well-known recursive formula for Lanczos iterations; the experimental condition number is computed as $\kappa_2 = \lambda_M/\lambda_m$.

The parallel code used to obtain the numerical results has been developed in Fortran and C; the Message Passing Interface (MPI) library has been used for paral-

lelization, assigning one subdomain to one MPI process. The BDDC preconditioner has been developed using the Portable Extensible Toolkit for Scientific Computation [5] (PETSc) and it is available for download within the development version of the library (see https://bitbucket.org/petsc/petsc). Whenever the BDDC algorithm is exactly applied, local problems are solved using the Unsymmetric Multifrontal sparse LU factorization package [7] (UMFPACK), while the algebraic multigrid (AMG) method boomerAMG provided by the HYPRE library [3] is used as a black-box solver within the inexact BDDC algorithm. The interested reader is referred to [13, 14] where the AMG method has been successfully applied to the serial and parallel solution of the Bidomain linear system. The BDDC coarse problem is solved in parallel either with the MUltifrontal Massively Parallel sparse direct Solver [4] (MUMPS) or inexaclty with the parallel boomerAMG method. For all test cases considered, the coarse space is spanned by subdomain vertices and edge averages for both cardiac potentials; unless otherwise stated, the conductivity coefficients used are reported in [6]. One $V_{1,1}$-cycle with Gauss-Seidel smoothing is always used for the AMG method in order to preserve symmetry of the resulting operator.

Table 1 contains results of a quasi-optimality test obtained on the x86_64 Linux cluster Matrix of CASPUR [1], where each core is equipped with 2GB memory. In this test case, Ω is divided in 3x3x3 subdomains, h=1E-2, δ_t=1E-2 and increasing values of H are considered; thus, the volume of Ω increases as H/h increases. Inexact solvers are used for both sets of local problems whereas the coarse problem is solved exactly with a parallel factorization. AMG based local solvers does not make the performances of the BDDC deteriorate with respect to H/h and they allow us to manage larger local problems, since the memory requirements for a multigrid preconditioner are linear in the local size. Quasi-optimality is thus preserved by the inexact BDDC algorithm for the Bidomain model.

Table 1 Comparison between exact and inexact BDDC method for different values of H/h. For each run, extreme eigenvalues, condition number and number of iterations are shown. Test case with h=1E-2 and 3x3x3 subdomains.

	$M_{BDDC}^{-1}\widehat{K}$				$\widetilde{M}_{BDDC}^{-1}\widehat{K}$			
$\frac{H}{h}$	λ_m	λ_M	κ_2	it	λ_m	λ_M	κ_2	it
5	1.00	1.45	1.45	6	0.88	1.42	1.61	7
10	1.00	2.28	1.28	9	0.88	2.14	2.45	10
15	1.00	2.98	2.98	11	0.87	2.66	3.06	11
20	1.00	3.49	3.49	13	0.87	3.17	3.71	13
25	1.00	4.02	4.02	13	0.85	3.56	4.18	14
30	*out of memory*				0.76	3.91	5.14	15
35	*out of memory*				0.75	4.23	5.60	16
40	*out of memory*				0.70	4.43	6.27	16

Table 2 contains experimental results of a weak scalability test for the inexact BDDC algorithm on the BlueGene/Q FERMI of CINECA [2]; total number of de-

grees of freedom (dofs), condition number, number of PCG iterations and solving time per iteration (time/it) in seconds are reported. In the test case, h=1E-2, H/h=30, δ_t=1E-2 and the number of subdomains N grows in each dimension as reported in the first two columns. Thus, the volume of Ω increases as N increases. Inexact solvers for both local problems and, in parallel, for the coarse problem are used. Results are scalable in the number of iterations and solving time per iteration up to 4K cores and 200 millions degrees of freedom.

Table 2 Weak scalability test for the inexact BDDC method. For each run, number of subdomains and domain decomposition, number of degrees of freedom (dofs), condition number, number of PCG iterations (it) and solving time per iteration are shown. Test case with h=1E-2 and H/h=30.

N	subd	dofs	$\kappa_2(\widetilde{M}^{-1}_{BDDC}\widehat{K})$	it	time/it (s)
8	2x2x2	410.758	5.79	13	0.96
64	4x4x4	3.203.226	5.79	13	0.94
512	8x8x8	25.298.674	9.81	15	1.01
4096	16x16x16	201.089.250	11.12	16	1.12

Finally, we report on a test case with coefficients with jumps aligned with Γ, obtained on the x86_64 Linux cluster Matrix of CASPUR [1]. As test case, we consider a 3x3x3 decomposition of Ω, h=1E-2, H/h=15 and δ_t=1E-2; inexact solvers are used for both local problems, instead the coarse problem is solved exactly with a parallel factorization. Two different checkerboard patterns of discontinuities in the conductivity coefficients are considered; conductivity coefficients are initially set to $\sigma_1^{i,e}$ =10, $\sigma_2^{i,e}$ =1 and $\sigma_3^{i,e}$ =0.1, then the following cases are built given a factor $p > 0$:

A Each conductivity coefficient, either intra- or extra-cellular, is multiplied by p in the black subdomains and by $1/p$ in the white subdomains.

B Intra-cellular coefficients are multiplied by p in the black subdomains and by $1/p$ in white subdomains; conversely, extra-cellular coefficients are multiplied by $1/p$ in the black subdomains and by p in white subdomains.

Numerical results are summarized in Table 3, with columns labeled according to the previous classification. The condition number and the number of iterations (listed in round brackets) of the inexact BDDC algorithm remain almost constant when we vary the factor p largely in both test cases considered; the ratio between inexact and exact condition number is also shown to highlight the quality of the inexact approach.

References

1. CASPUR HPC home page. http://hpc.caspur.it

Table 3 Condition number dependence of inexact BDDC method with coefficient jumps. For each run, condition number and number of iterations in round brackets are shown together with the ratio between condition numbers of the exact and inexact BDDC. Test case with h=1E-2, H/h=15 and 3x3x3 subdomains.

	A		**B**	
p	$\kappa_2(\widetilde{M}^{-1}_{BDDC}\widehat{K})$	ratio	$\kappa_2(\widetilde{M}^{-1}_{BDDC}\widehat{K})$	ratio
1	10.47 (20)	1.47	10.47 (20)	1.47
1E1	12.41 (22)	1.46	12.12 (21)	1.49
1E2	12.54 (22)	1.46	13.70 (24)	1.60
1E3	13.75 (23)	1.57	15.13 (24)	1.78

2. CINECA HPC home page. http://hpc.cineca.it
3. HYPRE. High performance preconditioners. http://www.llnl.gov/CASC/linear solvers/
4. Amestoy, P.R., Duff, I.S., Koster, J., L'Excellent, J.Y.: A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM J. Matrix Anal. Appl. **23**, 15–41 (2001)
5. Balay, S., Brown, J., Buschelman, K., Gropp, W.D., Kaushik, D., Knepley, M.G., Curfman McInnes, L., Smith, B.F., H., Z.: PETSc Web page (2012). Http://www.mcs.anl.gov/petsc
6. Colli Franzone, P., Pavarino, L.F.: A parallel solver for reaction diffusion systems in computational electrocardiology. Math. Mod. Meth. Appl. Sci. **14**, 883–911 (2004)
7. Davis, T.A.: Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. ACM Trans. Math. Soft. **30**, 196–199 (2004)
8. Dohrmann, C.R.: An approximate BDDC preconditioner. Numer. Lin. Alg. Appl. **14**, 149–168 (2007)
9. LeGrice, I.J., Smaill, B.H., Chai, L.Z., Edgar, S.G., Gavin, J.B., Hunter, P.J.: Laminar structure of the heart: ventricular myocyte arrangement and connective tissue architecture in the dog. Am. J. Physiol. Heart Circ. Physiol. **269**, H571–H582 (1995)
10. Li, J., Widlund, O.B.: FETI-DP, BDDC, and block Cholesky methods. Int. J. Numer. Meth. Engrg. **66**, 250–271 (2006)
11. Pavarino, L.F., Scacchi, S.: Multilevel additive Schwarz preconditioners for the Bidomain reaction-diffusion system. SIAM J. Sci. Comput. **31**, 420–443 (2008)
12. Pavarino, L.F., Scacchi, S.: Parallel multilevel Schwarz and block preconditioners for the Bidomain parabolic-parabolic and parabolic-elliptic formulations. SIAM J. Sci. Comput. **33**, 1897–1919 (2011)
13. Pennacchio, M., Simoncini, V.: Algebraic multigrid preconditioners for the Bidomain reaction-diffusion system. Appl. Numer. Math. **59**, 3033–3050 (2009)
14. Plank, G., Liebmann, M., Weber dos Santos, R., Vigmond, E., Haase, G.: Algebraic Multigrid Preconditioner for the cardiac Bidomain model. IEEE Trans. Biomed. Engrg. **54**, 585–596 (2007)
15. Scacchi, S.: A hybrid multilevel Schwarz method for the Bidomain model. Comp. Meth. Appl. Mech. Engrg. **197**, 4051–4061 (2008)
16. Scacchi, S.: A multilevel hybrid Newton-Krylov-Schwarz method for the Bidomain model of electrocardiology. Comp. Meth. Appl. Mech. Engrg. **200**, 717–725 (2011)
17. Toselli, A., Widlund, O.B.: Domain decomposition methods - Algorithms and theory. Springer (2005)
18. Vigmond, E., Weber dos Santos, R., Prassl, A.J., Deo, M., Plank, G.: Solvers for the cardiac Bidomain equations. Progr. Biophys. Mol. Biol. **96**, 3–18 (2008)
19. Zampini, S.: Balancing Neumann-Neumann methods for the cardiac Bidomain model. Numer. Math. **123**, 341–373 (2013)
20. Zampini, S.: Dual-Primal methods for the cardiac Bidomain model. Math. Models Methods Appl. Sci. (2013). DOI 10.1142/S0218202513500632

Parallel coupled and uncoupled multilevel solvers for the Bidomain model of electrocardiology

Piero Colli Franzone[1], Luca F. Pavarino[2], and Simone Scacchi[2]

1 Introduction

The Bidomain model describes the spread of electrical excitation in the anisotropic cardiac tissue in terms of the evolution of the transmembrane and extracellular electric potentials, v and u_e respectively. This model consists of a non-linear parabolic reaction-diffusion partial differential equation (PDE) for v, coupled with an elliptic linear PDE for u_e. The evolution equation is coupled through the non-linear reaction term with a stiff system of ordinary differential equations (ODEs), the so-called membrane model, describing the ionic currents through the cellular membrane. The different space and time scales involved make the solution of the Bidomain system a very challenging computational problem, because its discretization in three-dimensional ventricular geometries of realistic size requires the solution of large scale (often exceeding $O(10^7)$ unknowns) and ill-conditioned linear systems at each time step.

Several approaches have been developed in order to reduce the high computational costs of the Bidomain model. Fully implicit methods in time, requiring the solution of non-linear systems at each time step, have been considered in e.g. [10, 9]. Alternatively, most previous works have considered IMEX time discretizations and/or operator splitting schemes, where the reaction and diffusion terms are treated separately, see e.g. [2, 3, 18, 20, 23]. The advantage of IMEX and operator splitting schemes is that they only require the solution of a linear system for the parabolic and elliptic PDEs at each time step. A further splitting approach consists in uncoupling the parabolic PDE from the elliptic one, see e.g. [23, 4].

Many different preconditioners have been proposed in order to obtain efficient iterative solvers for the linear systems deriving from both splitting and uncoupling techniques: block diagonal or triangular [13, 14, 2, 22, 5], optimized Schwarz [6], multigrid [19, 16, 15, 13, 14], multilevel Schwarz [11], Balancing Neumann-Neumann [24] and BDDC [25] preconditioners.

The aim of the present work is to apply the Multilevel Additive Schwarz preconditioners of [11] to both a coupled and an uncoupled time discretization of the Bidomain system and to compare their parallel performance. Three-dimensional parallel numerical tests on a BlueGene cluster, reported in Sec. 4, show that the uncou-

[1] Department of Mathematics, University of Pavia, via Ferrata 1, 27100 Pavia, Italy, e-mail: colli@imati.cnr.it [2] Department of Mathematics, University of Milano, via Saldini 50, 20133 Milano, Italy, e-mail: {luca.pavarino}{simone.scacchi}@unimi.it

pled technique is as scalable as the coupled one. Moreover, the conjugate gradient method preconditioned by Multilevel Additive Schwarz preconditioners converges faster for the uncoupled system than for the coupled one. Finally, in all parallel numerical tests considered, the uncoupled technique proposed is always about 1.5 times faster than the coupled approach.

2 The anisotropic Bidomain model

The macroscopic Bidomain representation of the cardiac tissue volume Ω is obtained by considering the superposition of two anisotropic continuous media, the intra- (i) and extra- (e) cellular media, coexisting at every point of the tissue and separated by a distributed continuous cellular membrane; see e.g. [12] for a derivation of the Bidomain model from homogenization of cellular models. We recall that the cardiac tissue consists of an arrangement of fibers that rotate counterclockwise from epi- to endocardium, and that have a laminar organization modeled as a set of muscle sheets running radially from epi- to endocardium, see [7]. The anisotropy of the intra- and extracellular media is described by the orthotropic conductivity tensors $D_i(\mathbf{x})$ and $D_e(\mathbf{x})$, see e.g. [2].

We denote by $\Omega \subset \mathbb{R}^3$ the bounded physical region occupied by the cardiac tissue and introduce a parabolic-elliptic formulation of the Bidomain system. Given an applied extracellular current per unit volume $I^e_{app} : \Omega \times (0,T) \to \mathbb{R}$, we seek the transmembrane potential $v : \Omega \times (0,T) \to \mathbb{R}$, extracellular potentials $u_e : \Omega \times (0,T) \to \mathbb{R}$, gating variables $w : \Omega \times (0,T) \to \mathbb{R}^{N_w}$ and ionic concentrations $c : \Omega \times (0,T) \to \mathbb{R}^{N_c}$ such that

$$\begin{cases} c_m \dfrac{\partial v}{\partial t} - \mathrm{div}(D_i(\mathbf{x})\nabla v) - \mathrm{div}(D_i(\mathbf{x})\nabla u_e) + I_{ion}(v,w,c) = 0 & \text{in } \Omega \times (0,T) \\ -\mathrm{div}(D_i(\mathbf{x})\nabla v) - \mathrm{div}((D_i(\mathbf{x}) + D_e(\mathbf{x}))\nabla u_e) = I^e_{app} & \text{in } \Omega \times (0,T) \\ \dfrac{\partial w}{\partial t} - R(v,w) = 0, \ \dfrac{\partial c}{\partial t} - S(v,w,c) = 0, & \text{in } \Omega \times (0,T) \end{cases} \quad (1)$$

with insulating boundary conditions, suitable initial conditions on v, w, c and where c_m is the membrane capacitance per unit volume. The non-linear reaction term I_{ion} and the ODE system for the gating variables w and the ionic concentrations c are given by the chosen ionic membrane model. Here we will consider the Luo-Rudy I (LR1) membrane model [8].

3 Discretization and numerical methods

Space discretization. The variational formulation of system (1) is first discretized in space by the finite element method. In this work, we will consider isoparametric

trilinear finite elements on hexahedral meshes. In the following, we denote by $A_{i,e}$ the symmetric intra- and extracellular stiffness matrices and by M the mass matrix. We define the block mass and stiffness matrices as

$$\mathbb{M} = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}, \qquad \mathbb{A} = \begin{bmatrix} A_i & A_i \\ A_i & A_i + A_e \end{bmatrix}.$$

Time discretization. We consider two implicit-explicit (IMEX) strategies, both based on decoupling the ODEs from the PDEs and on treating the linear diffusion terms implicitly and the non-linear reaction terms explicitly.

- **Coupled method.** The equations arising from the discretization of the PDEs are solved as a coupled system. Given $\mathbf{w}^n$, $\mathbf{c}^n$, $\mathbf{v}^n$, $\mathbf{u}_e^n$ at the generic time step n:
  - we first solve the ODEs system using the Implicit Euler method for the gating variables and the Explicit Euler method for the ionic concentrations, obtaining the new gating variables $\mathbf{w}^{n+1}$ and the new ionic concentrations $\mathbf{c}^{n+1}$,
  - then we solve the PDEs system, obtaining the new potentials $\mathbf{v}^{n+1}$ and $\mathbf{u}_e^{n+1}$.

  Summarizing in formulae, given $\mathbf{w}^n$, $\mathbf{c}^n$, $\mathbf{v}^n, \mathbf{u}_e^n$, the scheme is

$$\begin{aligned} \mathbf{w}^{n+1} - \Delta t\, \mathbf{R}(\mathbf{v}^n, \mathbf{w}^{n+1}) &= \mathbf{w}^n \\ \mathbf{c}^{n+1} &= \mathbf{c}^n + \Delta t\, \mathbf{S}(\mathbf{v}^n, \mathbf{w}^{n+1}, \mathbf{c}^n) \\ \left(\frac{c_m}{\Delta t}\mathbb{M} + \mathbb{A}\right) \begin{bmatrix} \mathbf{v}^{n+1} \\ \mathbf{u}_e^{n+1} \end{bmatrix} &= \frac{c_m}{\Delta t}\mathbb{M} \begin{bmatrix} \mathbf{v}^n \\ \mathbf{u}_e^n \end{bmatrix} + \begin{bmatrix} -M\mathbf{I}_{ion}(\mathbf{v}^n, \mathbf{w}^{n+1}, \mathbf{c}^{n+1}) \\ M\mathbf{I}_{app}^{e,n+1} \end{bmatrix} \end{aligned}.$$

  As a consequence, at each time step, we solve one linear system with unknowns $(\mathbf{v}^{n+1}, \mathbf{u}_e^{n+1})$. Because the iteration matrix is symmetric positive semi-definite, the iterative method employed is the preconditioned conjugate gradient (PCG) method. Due to the ill-conditioning of the iteration matrix and the large number of unknowns required by realistic simulations of cardiac excitation in three-dimensional domains, a scalable and efficient preconditioner is required. We adopt here the 4-level Multilevel Additive Schwarz (MAS(4)) preconditioner, see [21, 11].
- **Uncoupled method**. The two equations arising from the discretization of the PDEs are uncoupled by introducing the following scheme. Given $\mathbf{w}^n$, $\mathbf{c}^n$, $\mathbf{v}^n$, $\mathbf{u}_e^n$ at the generic time step n:
  - we first solve the ODEs system using the Implicit Euler method for the gating variables and the Explicit Euler method for the ionic concentrations, obtaining the new gating variables $\mathbf{w}^{n+1}$ and the new ionic concentrations $\mathbf{c}^{n+1}$,
  - then we solve the elliptic equation, obtaining $\mathbf{u}_e^n$,
  - and finally we update the transmembrane potential $\mathbf{v}^{n+1}$ by solving again the parabolic equation.

  Summarizing in formulae, given $\mathbf{w}^n$, $\mathbf{c}^n$, $\mathbf{v}^n, \mathbf{u}_e^n$, the uncoupled scheme is

$$\begin{aligned}
\mathbf{w}^{n+1} - \Delta t\ \mathbf{R}(\mathbf{v}^n, \mathbf{w}^{n+1}) &= \mathbf{w}^n \\
\mathbf{c}^{n+1} &= \mathbf{c}^n + \Delta t\ \mathbf{S}(\mathbf{v}^n, \mathbf{w}^{n+1}, \mathbf{c}^n) \\
(A_i + A_e)\mathbf{u}_e^n &= -A_i \mathbf{v}^n + M\mathbf{I}_{app}^{e,n} \\
\left(\frac{c_m}{\Delta t} M + A_i\right)\mathbf{v}^{n+1} &= \frac{c_m}{\Delta t} M\mathbf{v}^n - A_i \mathbf{u}_e^n - M\mathbf{I}_{ion}(\mathbf{v}^n, \mathbf{w}^{n+1}, \mathbf{c}^{n+1}).
\end{aligned}$$

As a consequence, at each time step we solve first the linear system with matrix $A_i + A_e$ deriving from the elliptic equation and afterwards the linear system with matrix $\frac{c_m}{\Delta t} M + A_i$ deriving from the parabolic equation. Both linear systems are solved by the PCG method, since the matrices are symmetric positive definite in the parabolic case and semi-definite in the elliptic case. The preconditioner used for the parabolic system is Block Jacobi (BJ), because the related matrix is well-conditioned, while the preconditioner used for the ill-conditioned elliptic system is the MAS(4) preconditioner, described below.

Multilevel Additive Schwarz preconditioners. Let Ω^k, for $k = 0, \ldots, \ell - 1$, be a family of ℓ nested triangulations of Ω, coarsening from $\ell - 1$ to 0, $A^{\ell-1} = \mathbb{A}$ in the coupled method and $A^{\ell-1} = A_i + A_e$ in the uncoupled method, and R^k the restriction operators from $\Omega^{\ell-1}$ to Ω^k. Define the matrices on each grid as $A^k = R^k A^{\ell-1} R^{k^T}$ for $k = 0, \ldots, \ell - 2$. We then decompose each grid Ω^k, for $k = 1, \ldots, \ell - 1$, into N overlapping subgrids Ω_m^k for $m = 1, \ldots, N$ and define the local restriction operators R_m^k from $\Omega^{\ell-1}$ to Ω_m^k and the local matrices $A_m^k = R_m^k A^{\ell-1} R_m^{k^T}$. The Multilevel Additive Schwarz (MAS(ℓ)) preconditioner is given by

$$B_{MAS}^{-1} = R^{0^T} A^{0^{-1}} R^0 + \sum_{k=1}^{\ell-1} \sum_{m=1}^{N} R_m^{k^T} A_m^{k^{-1}} R_m^k.$$

The condition number of the resulting preconditioner operator $T_{MAS} = B_{MAS}^{-1} A^{\ell-1}$ is bounded by

$$\kappa_2(T_{MAS}) \leq C \max_{k=1,\ldots,\ell-1} \left(1 + \frac{h_{k-1}}{\delta_k}\right),$$

where h_k is the mesh size of Ω^k grid, δ_k is the overlap size on level k and C is a constant independent of h_k, δ_k, N and ℓ; see [11] and for hybrid variants [17].

4 Numerical results

In this section, we present the results of parallel numerical experiments performed on the BlueGene Cluster BG/Q of the Cineca Consortium (www.cineca.it). Our FORTRAN code is based on the parallel library PETSc [1], from the Argonne National Laboratory.

procs	*dof*	coupled			uncoupled		
		$\kappa_2 = \lambda_M/\lambda_m$	*it*	*time*	$\kappa_2 = \lambda_M/\lambda_m$	*it*	*time*
64	4,319,890	41.85=8.70/2.08e-1	43	5.65	15.52=4.50/2.90e-1	29	1.82+1.07=2.89
128	8,553,474	33.41=6.79/2.03e-1	39	5.57	14.94=4.46/2.99e-1	28	2.02+1.03=3.05
256	17,040,642	36.37=6.81/1.87e-1	40	5.70	15.36=4.46/2.91e-1	28	1.92+1.05=2.97
512	33,949,186	27.37=5.16/1.88e-1	36	5.48	14.35=4.38/3.05e-1	28	1.98+0.99=2.97
1,024	67,766,274	29.53=5.16/1.75e-1	36	5.69	14.43=4.42/3.06e-1	28	2.17+1.04=3.21
2,048	135,268,866	27.56=5.08/1.84e-1	34	8.50	13.23=4.33/3.28e-1	27	2.93+1.72=4.65
4,096	270,274,050	28.91=5.09/1.76e-1	34	16.39	13.23=4.33/3.28e-1	27	5.58+3.63=9.21
8,192	540,021,250	25.03=5.10/2.04e-1	32	16.51	12.41=4.30/3.47e-1	26	5.93+3.75=9.68
16,384	1,079,515,650	26.55=5.11/1.92e-1	32	17.39	12.41=4.30/3.47e-1	26	6.24+3.83=10.07
32,768	2,159,978,114	–			12.03=4.32/3.59e-1	26	6.90+3.94=10.84

Table 1 Test 1. Weak scaling for coupled and uncoupled MAS(4) solvers on ellipsoidal structured meshes. Average condition number (κ_2), extreme eigenvalues (λ_M, λ_m), PCG iteration count (*it*) and CPU time in seconds (*time*) per time step. The CPU times in the uncoupled column are expressed as the sum of the elliptic plus the parabolic solver. The run with 32K cores in the case of coupled solver failed because of RAM limitations.

4.1 Test 1: weak scaling on ellipsoidal domains, structured mesh

The coupled and uncoupled linear solvers are compared here in a scaled speedup test on ellipsoidal deformed domains, discretized by structured Q_1 finite element grids. The number of subdomains (and processors) is increased from 64 to 32,768, forming increasing ellipsoidal domains Ω. The fine mesh is chosen so as to keep the local mesh size on each subdomain fixed at $32 \times 32 \times 32$. With these choices, the global size of the discrete Bidomain system increases from about 4 million dof for the smallest domain with 64 subdomains to more than 2 billion dof for the largest domain with 32,768 subdomains. The physical dimensions of the increasing cartesian slabs are chosen so that the fine mesh size h is kept fixed to the value $h = 0.01$ *cm*. The simulation is run for 10 time steps of 0.05 *ms* during the depolarization phase, which is the most intense computationally.

The results reported in Table 1 clearly show that, since the MAS(4) preconditioner is employed, both the coupled and uncoupled methods are scalable. In fact, all mathematical quantities (condition number, extreme eigenvalues, PCG iteration count) seem to approach constant values when increasing the number of subdomains. Also the CPU times scale quite well, because they only increase of about a factor $3-4$ from 64 to 32,768 processors, with a very small and slow increase after 4096, while the global problem increases by a factor 512.

method	*it*	*Tit*	*time*	*Ttime*
coupled	22	82,861	2.43	9.29e+3
uncoupled	27	92,157	1.72	5.87e+3

Table 2 Test 2. Comparison of coupled and uncoupled solvers on a whole heartbeat simulation, with $28,755,650$ dof, $1,024$ processors. Average PCG iteration count (*it*) and CPU time (*time*) per time step, total PCG iteration count (*Tit*) and CPU time (*Ttime*). The CPU times are expressed in seconds.

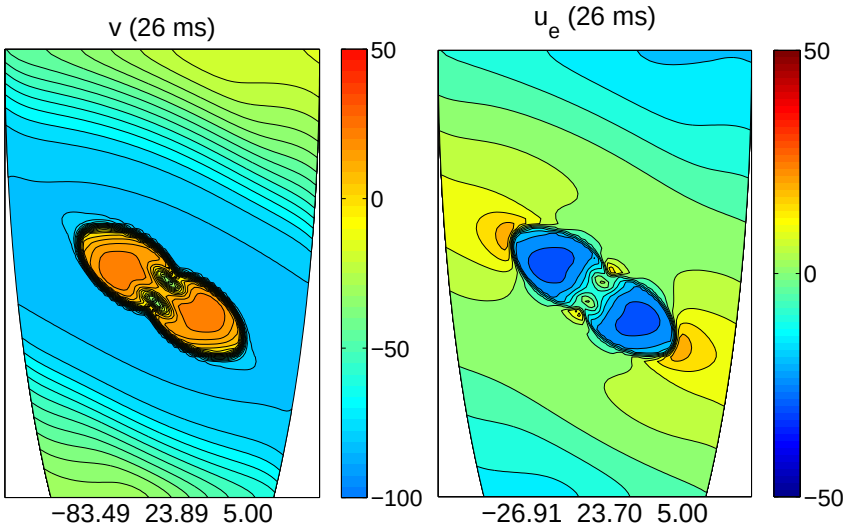


Fig. 1 Test 2. Epicardial transmembrane (left) and extracellular (right) potential distributions at $t = 26$ *ms* after an electric stimulus applied during the systolic phase of the heart beat.

4.2 Test 2: comparison between coupled and uncoupled methods on a complete cardiac cycle simulation

We now compare the coupled and uncoupled solvers on a complete heartbeat (500 *ms*) in a portion of an ellipsoid, modeling half of the left ventricle, discretized by a Q_1 structured finite element grid of $384 \times 384 \times 96$ elements ($28,755,650\ dof$). The MAS(4) preconditioner is employed in the coupled solver and for the elliptic linear system in the uncoupled solver, while the BJ preconditioner is employed for the parabolic linear system in the uncoupled solver. The simulations are run on 1,024 cores. The time step size is changed according to the adaptive strategy described in [2].

The results reported in Table 2 show that the uncoupled method is about 1.5 times faster than the coupled one, because at each time step one solves two linear system of half size, the parabolic one being well conditioned and cheap to solve.

Fig. 1 reports the epicardial transmembrane and extracellular potential distributions at $t = 26$ *ms* after an electric stimulus has been applied during the systolic phase of the heart beat at the center of the epicardial surface.

5 Conclusion

We have applied Multilevel Additive Schwarz preconditioners to both coupled and uncoupled time discretizations of the Bidomain model of the cardiac bioelectric activity and we have compared their parallel performance. Three-dimensional parallel numerical tests on a BlueGene/Q cluster up to 32K cores have shown that the uncoupled technique is as scalable as the coupled one. Moreover, the conjugate gradient method preconditioned by Multilevel Additive Schwarz preconditioners converges faster for the uncoupled system than for the coupled one. Finally, in all parallel numerical tests considered, the uncoupled technique proposed was always about 1.5 times faster than the coupled approach.

References

1. Balay, S., et al.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.3, Argonne National Laboratory (2012)
2. Colli Franzone, P., Pavarino, L.F.: A parallel solver for reaction-diffusion systems in computational electrocardiology. Math. Mod. Meth. Appl. Sci. **14**, 883–911 (2004)
3. Ethier, M., Bourgault, Y.: Semi-implicit time-discretization schemes for the bidomain model. SIAM J. Numer. Anal. **46**, 2443–2468 (2008)
4. Fernandez, M.A., Zemzemi, N.: Decoupled time-marching schemes in computational cardiac electrophysiology. Math. Biosci. **226**, 58–75 (2010)
5. Gerardo Giorda, L., et al.: A model-based block-triangular preconditioner for the bidomain system in electrocardiology. J. Comp. Phys. **228**, 3625–3639 (2009)
6. Gerardo Giorda, L., et al.: Optimized Schwarz coupling of bidomain and monodomain models in electrocardiology. Math. Model. Numer. Anal. **45**, 309–334 (2011)
7. LeGrice, I.J., et al.: Laminar structure of the heart: ventricular myocyte arrangement and connective tissue architecture in the dog. Am. J. Physiol. Heart Circ. Physiol. **269**, H571–H582 (1995)
8. Luo, C., Rudy, Y.: A model of the ventricular cardiac action potential: depolarization, repolarization, and their interaction. Circ. Res. **68**, 1501–1526 (1991)
9. Munteanu, M., Pavarino, L.F., Scacchi, S.: A scalable Newton-Krylov-Schwarz method for the bidomain reaction-diffusion system. SIAM J. Sci. Comput. **31**, 3861–3883 (2009)
10. Murillo, M., Cai, X.C.: A fully implicit parallel algorithm for simulating the non-linear electrical activity of the heart. Numer. Linear Algebra Appl. **11**, 261–277 (2004)
11. Pavarino, L.F., Scacchi, S.: Multilevel additive Schwarz preconditioners for the Bidomain reaction-diffusion system. SIAM J. Sci. Comput. **31**, 420–443 (2008)
12. Pennacchio, M., Savaré, G., Colli Franzone, P.: Multiscale modeling for the bioelectric activity of the heart. SIAM J. Math. Anal. **37**, 1333–1370 (2006)
13. Pennacchio, M., Simoncini, V.: Algebraic multigrid preconditioners for the bidomain reaction-diffusion system. Appl. Numer. Math. **59**, 3033–3050 (2009)
14. Pennacchio, M., Simoncini, V.: Fast structured AMG preconditioning for the bidomain model in electrocardiology. SIAM J. Sci. Comput. **33**, 721–745 (2011)

15. Plank, G., et al.: Algebraic multigrid preconditioner for the cardiac bidomain model. IEEE Trans. Biomed. Engrg. **54**, 585–596 (2007)
16. Weber dos Santos, R., et al.: Parallel multigrid preconditioner for the cardiac bidomain model. IEEE Trans. Biomed. Engrg. **51**, 1960–1968 (2004)
17. Scacchi, S.: A hybrid multilevel Schwarz method for the bidomain model. Comput. Meth. Appl. Mech. Engrg. **197**, 4051–4061 (2008)
18. Southern, J.A., et al.: Solving the coupled system improves computational efficiency of the bidomain equations. IEEE Trans. Biomed. Engrg. **56**, 2404–2412 (2009)
19. Sundnes, J., et al.: Multigrid block preconditioning for a coupled system of partial differential equations modeling the electrical activity in the heart. Comput. Meth. Biomech. Biomed. Engrg. **5**, 397–409 (2002)
20. Sundnes, J., Lines, G.T., Tveito, A.: An operator splitting method for solving the bidomain equations coupled to a volume conductor model for the torso. Math. Biosci. **194**, 233–248 (2005)
21. Toselli, A., Widlund, O.B.: Domain Decomposition Methods - Algorithms and Theory. Springer-Verlag, Berlin (2004)
22. Vigmond, E.J., Aguel, F., Trayanova, N.A.: Computational techniques for solving the bidomain equations in three dimensions. IEEE Trans. Biomed. Engrg. **49**, 1260–1269 (2002)
23. Vigmond, E.J., et al.: Solvers for the cardiac bidomain equations. Progr. Biophys. Molec. Biol. **96**, 3–18 (2008)
24. Zampini, S.: Balancing Neumann-Neumann methods for the cardiac bidomain model. Numer. Math. **123**, 363–393 (2013)
25. Zampini, S.: Inexact BDDC methods for the cardiac bidomain model. In: Proceedings of DD21 (2013)

Fuzzy Domain Decomposition: a new perspective on heterogeneous DD methods

Martin J. Gander[1] and Jérôme Michaud[1]

1 Motivation

In a wide variety of physical problems, the complexity of the physics involved is such that it is necessary to develop approximations, because the complete physical model is simply too costly. Sometimes however the complete model is essential to capture all the physics, and often this is only in part of the domain of interest. One can then use heterogeneous domain decomposition techniques: if we know a priori where an approximation is valid, we can divide the computational domain into subdomains in which a particular approximation is valid and the topic of heterogeneous domain decomposition methods is to find the corresponding coupling conditions to insure that the overall coupled solution is a good approximation of the solution of the complete physical model. For an overview of such techniques, see [9, 10] and references therein. However, there are many physical problems where it is not a priori known where which approximation is valid. In such problems, one needs to track the domain of validity of a particular approximation, and this is usually not an easy task. An example of such a method is the χ-method, see [4, 1].

In this contribution, we introduce a new formalism for heterogeneous domain decomposition, which is not based on a sharp decomposition into subdomains where different models are valid. The main idea relies on the notion of *Fuzzy Sets* introduced by Zadeh [12] in 1965. The Fuzzy Set Theory relaxes the notion of belonging to a set through *membership functions* to (fuzzy) sets that account for partially belonging to a set. In the context of heterogeneous domain decomposition, this could be useful if one assumes that the computational domain can be decomposed into fuzzy sets that form a partition of the domain in a sense that needs to be specified. Once such a partition is given, one can compute the solution of the coupled problem using the membership functions. Note that the membership functions can depend on space and time and therefore can take into account a change in the validity domain of a particular approximation. We show here that this technique leads to an excellent coupling strategy for the 1D advection dominated diffusion problem. Such a domain decomposition method would be able, in principle, to take into account part of the domain where none of the available approximations are valid under the assumption that a combination of them is a good enough approximation there.

On the assumption $\mathbf{u} = \mathbf{u_1} + \mathbf{u_2}$: The idea to use fuzzy set theory came from an assumption that arose in some specific coupling methods (see below). We formulate it here for a generic partial differential equation of the form

[1]Université de Genève, 2-4 rue du Lièvre, CP 64, CH-1211 Genève 4,
e-mail: {Martin.Gander}{Jerome.Michaud}@unige.ch

$$\mathscr{L}(u) = g, \tag{1}$$

where $\mathscr{L}$ is a linear differential operator.

Assumption 1 *($u = u_1 + u_2$).We assume that the solution u of (1) can be written as a sum, $u = u_1 + u_2$, and that one can derive a coupled system for the new unknowns u_1 and u_2. The derivation of the coupled system might then use relevant approximations for one or both components.*

This assumption has been used at least in two different series of papers: the first one is in physics for the approximation of neutrino radiative transfer in core-collapse supernovae [11, 2, 3], and the second one is in mathematics for the coupling between the kinetic equation and approximations of it (diffusion, Euler, Navier-Stokes...) [8, 5, 6, 7].

In the following, we will see how this assumption can be linked with fuzzy sets. This will lead us to introduce fuzzy domain decomposition methods.

2 Fuzzy Sets and Fuzzy Domain Decomposition Methods

Let X be a set in the classical sense of generic elements x, such that $X = \{x\}$.

Definition 1 (Fuzzy Set). A *fuzzy set* A of X is characterized by a *membership function* $h_A(x)$ that associates to every point of X a real number in $[0,1]$. The value of $h_A(x)$ represents the *grade of membership of x in A*. The *support* $\mathrm{Supp}(A)$ of a fuzzy set A is the classical subset of X defined by $\mathrm{Supp}(A) = \{x \in X | h_A(x) \neq 0\}$.

Remark 1. If the membership function is a characteristic function, then we recover the classical notion of sets.

We next list a few useful properties of fuzzy sets:

Definition 2 (Complementary set). The *complementary set* A^c of a fuzzy set A is defined by its membership function $h_{A^c} = 1 - h_A$.

Definition 3 (Union of fuzzy sets). The *union* of two fuzzy sets A and B of membership function $h_A(x)$ and $h_B(x)$ is the fuzzy set C, denoted by $C = A \cup B$. It is characterized by its membership function $h_C(x)$ linked with those of A and B by $h_C(x) = \max(h_A(x), h_B(x))$, $\forall x \in X$.

Remark 2. The union of a fuzzy set with its complementary set is not equal to the initial set, unless the membership functions are characteristic functions: $A \cup A^c \subsetneq X$.

Definition 4 (Algebraic sum of fuzzy sets). The *algebraic sum* of A and B is denoted by $A + B$ and is defined by the membership function $h_{A+B} = h_A + h_B$. This definition has a meaning only if $h_A(x) + h_B(x) \leq 1$, $\forall x \in X$.

Remark 3. Note that the algebraic sum has the property that $A + A^c = X$.

Let Ω be the computational domain of the problem we want to solve. We use the *algebraical sum* of fuzzy sets to obtain a decomposition of the domain:

Definition 5 (Fuzzy Domain Decomposition (FDD)). A *fuzzy domain decomposition* is given by the fuzzy sets Ω_i, $i = 1, \ldots, n$ defined by their membership functions h_i such that their algebraic sum equals the domain Ω: $\Omega = \Omega_1 + \cdots + \Omega_n$. In terms of membership functions, this condition reads $\sum_{i=1}^{n} h_i(x) = 1$, $\forall x \in \Omega$.

Definition 6. Let u be a function from Ω to $\mathbb{R}$. We define the *restriction of u to the fuzzy set A* of Ω by $u_A = h_A u$, where h_A is the membership function of A.

Proposition 1. *Let u be a function from Ω to $\mathbb{R}$, let $\{\Omega_i\}_{i=1}^{n}$ be a fuzzy domain decomposition of Ω and let u_i be the restriction of u to Ω_i. Then*

$$u = \sum_{i=1}^{n} u_i \quad \textit{and} \quad u' = \sum_{i=1}^{n} u_i'. \tag{2}$$

Proof. This is a direct consequence of Definition 6 of the restriction of u to fuzzy sets, and the linearity of derivatives. □

Definition 7 (FDDM, eFDDM, iFDDM). A *FDD method* (FDDM), is a numerical method based on an FDD of the domain. We will say that an FDDM is explicit (eFDDM) if the membership functions h_i are explicitly known, and implicit otherwise (iFDDM).

Remark 4. The relation (2) shows that if the Assumption 1 is used, it is natural to interpret the resulting method as an FDDM. The methods of Degond et al. [8, 5, 6, 7] belong to the eFDDM class, but the IDSA [11, 2, 3] is an example of an iFDDM.

If we want to obtain an heterogeneous DDM, we need two ingredients. The first one is a coupling methodology between the two approximations (one of them may be exact), and the second one is a criterion to decide where an approximation is valid. The advantage of an eFDDM is that the h_i functions are used both for implementing the coupling and the criterion. As the partition is explicitly known, we can change it to test various criteria for the validity of the different approximations.

We now show the coupling procedure for a decomposition into two fuzzy domains. Assume that we want to solve an approximation of Problem (1) and that we have two approximations $\mathscr{L}_1$ and $\mathscr{L}_2$ of the linear operator $\mathscr{L}$ valid in a fuzzy sense in Ω_1 and Ω_2 respectively. Then, we can decompose Problem (1) as

$$\mathscr{L}(u^*) = g \quad \Leftrightarrow \quad h_1\mathscr{L}(u^*) + h_2\mathscr{L}(u^*) = g \quad \rightsquigarrow \quad h_1\mathscr{L}_1(u) + h_2\mathscr{L}_2(u) = g, \tag{3}$$

where we have introduced in the last formulation the approximated operators. Here, u^* stands for the exact solution and u for the approximate solution. The symbol $\rightsquigarrow$ means "is approximated by". In order to obtain a FDDM, we will use Assumption 1 , and to obtain an explicit method in the sense of Definition 7, we require

$$u_i = h_i u, \quad u_i' = h_i' u + h_i u', \quad u_i'' = h_i'' u + 2h_i' u' + h_i u'', \quad i = 1,2, \tag{4}$$

where we used the product rule for h_i sufficiently smooth.

As $g = h_1 g + h_2 g$, we can rewrite Equation $(3)_3$ as a system

$$\begin{cases} h_1 \mathscr{L}_1(u) = h_1 g \text{ on } \Omega, \\ h_2 \mathscr{L}_2(u) = h_2 g \text{ on } \Omega, \end{cases} \quad \rightsquigarrow \quad \begin{cases} \widetilde{\mathscr{L}_1}(u_1) = h_1 g + \mathscr{L}_{12}(u_2) \text{ on } \mathrm{Supp}(\Omega_1), \\ \widetilde{\mathscr{L}_2}(u_2) = h_2 g + \mathscr{L}_{21}(u_1) \text{ on } \mathrm{Supp}(\Omega_2). \end{cases} \tag{5}$$

The second system is obtained by using Assumption 1 and Equation (4). The use of the product rule to handle the fact that the h_i do not commute with $\mathscr{L}_i$ leads to the operators $\widetilde{\mathscr{L}_i}$ and $\mathscr{L}_{i,3-i}$ that are linked by the relation

$$\widetilde{\mathscr{L}_i} = \mathscr{L}_i - \mathscr{L}_{i,3-i}, \quad i = 1,2. \tag{6}$$

The change in support simply reflects the fact that Equation $(5)_1$ is non-trivial only in $\mathrm{Supp}(\Omega_i)$. Equation $(5)_2$ is an eFDDM for Problem $(3)_3$.

Remark 5. The boundary conditions of an eFDDM can be easily defined by transferring the boundary conditions on u to u_i using Equation (4).

3 An Example: Advection Dominated Diffusion

As an example, we consider for $\nu, a > 0$ the 1D advection diffusion equation

$$\mathscr{L}(u^*) = \nu u^{*\prime\prime} + a u^{*\prime} = 0 \quad \text{on } (0,1), \quad u^*(0) = 0, \quad u^*(1) = 1, \tag{7}$$

whose closed form solution is given by $u^*(x) = \frac{e^{-ax/\nu}-1}{e^{-a/\nu}-1}$. For $\frac{\nu}{a} \ll 1$, the diffusion term is only important close to 0 where a boundary layer forms. We can define the operators

$$\mathscr{L}_1 := \mathscr{L} = \nu \partial_{xx} + a\partial_x, \quad \text{and} \quad \mathscr{L}_2 := a\partial_x, \tag{8}$$

and, as before, using Assumption 1 and Equation (4) we have

$$\mathscr{L}_{12} := \nu(h_1'' + 2h_1'\partial_x) + a h_1' \quad \text{and} \quad \mathscr{L}_{21} := a h_2'. \tag{9}$$

The eFDDM method we get with the operators from (8,9), using Equation (6) to define $\widetilde{\mathscr{L}_i}$, with $g = 0$, is

$$\begin{aligned} \nu u_1'' + (a - 2\nu h_1')u_1' - (\nu h_1'' + a h_1')u_1 &= 2\nu h_1' u_2' + (\nu h_1'' + a h_1')u_2, && \text{on } \mathrm{Supp}(\Omega_1), \\ a u_2' - a h_2' u_2 &= a h_2' u_1, && \text{on } \mathrm{Supp}(\Omega_2). \end{aligned} \tag{10}$$

Under Assumption 1 and Equation (4), Equations $(5)_2$ and $(3)_3$ are equivalent.

The problem we are solving is then equivalent, by Equation $(3)_3$, to

$$h_1 \nu u'' + a u' = 0, \quad \text{on } (0,1), \quad u(0) = 0,\ u(1) = 1, \tag{11}$$

whose analytical solution, provided that $\mathrm{Supp}(\Omega_1)$ is connected, is given by

$$u(x) = \frac{\int_0^x (e^{-\int_0^y \frac{a}{\nu h_1(z)} dz}) dy}{\int_{\mathrm{Supp}(\Omega_1)} (e^{-\int_0^y \frac{a}{\nu h_1(z)} dz}) dy}, \text{ if } x \in \mathrm{Supp}(\Omega_1),\ u(x) = 1, \text{ otherwise.} \quad (12)$$

We now study the approximation quality of this method as $\frac{\nu}{a} \to 0$ for a decreasing twice continuously differentiable membership function h_1 of the form

$$h_1(x) := 1, \text{ if } 0 \le x \le c_1, \quad h_1(x) := h(x), \text{ if } c_1 < x < c_2, \quad h_1(x) := 0, \text{ if } c_2 \le x \le 1, \quad (13)$$

where $0 < h(x) \le 1$, so that $\mathrm{Supp}(\Omega_1)$ in Equation (12) is $\mathrm{Supp}(\Omega_1) = [0, c_2)$. We define $\delta := c_2 - c_1$ to be the width of the coupling region.

Theorem 1. *For h_1 as in Equation (13), the relative error $err_{App}(\frac{\nu}{a}) := \frac{\|u-u^*\|_{L^2(0,1)}}{\|u^*\|_{L^2(0,1)}}$ satisfies when $\frac{\nu}{a} \to 0$ the estimates:*

	$c_1 = cst.,$ $\delta = cst.$	$c_1 = \kappa \left(\frac{\nu}{a}\right)^{1-\varepsilon},$ $\delta = \kappa' \left(\frac{\nu}{a}\right)^{1-\varepsilon}$	$c_1 = \kappa \frac{\nu}{a} \ln(\frac{a}{\nu}),$ $\delta = \kappa' \frac{\nu}{a}$	$c_1 = \kappa \frac{\nu}{a},$ $\delta = \kappa' \frac{\nu}{a}$
$err_{App}(\frac{\nu}{a})$	$O(e^{-\frac{ac_1}{\nu}})$	$O\left(e^{-\kappa(\frac{a}{\nu})^{\varepsilon}}\right)$	$O(\ln(\frac{a}{\nu})^{0.5}(\frac{\nu}{a})^{\kappa+0.5})$	$O((\frac{\nu}{a})^{0.5})$

(14)

Here, $\kappa > 0$, $\kappa' \ge 0$ are constants, and $0 < \varepsilon \le 1$.

Proof. The proof of this result is divided into 3 steps. Step 1 finds two functions $\tilde{u}_1^*$ and $\tilde{u}_2^*$ that satisfy $\tilde{u}_1^* \le u \le \tilde{u}_2^*$. With such functions, we always have the bound

$$\frac{\|u-u^*\|_{L^2(0,1)}}{\|u^*\|_{L^2(0,1)}} \le \max_{i=1,2} e_i, \quad e_i := \frac{\|\tilde{u}_i^* - u^*\|_{L^2(0,1)}}{\|u^*\|_{L^2(0,1)}}. \quad (15)$$

Step 2 estimates $\max_{i=1,2} e_i^2$ and step 3 handles the 4 cases in (14).
Step 1: With h_1 as in Equation (13), we can express the function u as

$$u(x) = \begin{cases} \frac{1-e^{-\frac{ax}{\nu}}}{1-e^{-\frac{ac_1}{\nu}}\left(1-\frac{a}{\nu}\int_{c_1}^{c_2} e^{-\frac{a}{\nu}\int_{c_1}^{y} h^{-1}(z)dz} dy\right)}, & \text{if } 0 \le x \le c_1, \\ \frac{1-e^{-\frac{ac_1}{\nu}}\left(1-\frac{a}{\nu}\int_{c_1}^{x} e^{-\frac{a}{\nu}\int_{c_1}^{y} h^{-1}(z)dz} dy\right)}{1-e^{-\frac{ac_1}{\nu}}\left(1-\frac{a}{\nu}\int_{c_1}^{c_2} e^{-\frac{a}{\nu}\int_{c_1}^{y} h^{-1}(z)dz} dy\right)}, & \text{if } c_1 < x < c_2, \\ 1, & \text{if } c_2 \le x \le 1. \end{cases}$$

Using the fact that $0 < h(z) \le 1$, we have the estimate

$$1 - e^{-\frac{ac_1}{\nu}} < 1 - e^{-\frac{ac_1}{\nu}}\left(1 - \frac{a}{\nu}\int_{c_1}^{x} e^{-\frac{a}{\nu}\int_{c_1}^{y} h^{-1}(z)dz} dy\right) \le 1 - e^{-\frac{ax}{\nu}}, \quad c_1 < x < c_2.$$

Using this estimate, we define $\tilde{u}_i^*$, $i = 1, 2$ as

$$\left.\begin{array}{ll}\text{if } 0\le x\le c_1, & \frac{1-e^{-\frac{ax}{\nu}}}{1-e^{-\frac{ac_2}{\nu}}}\\ \text{if } c_1<x<c_2, & \frac{1-e^{-\frac{ac_1}{\nu}}}{1-e^{-\frac{ac_2}{\nu}}}\\ \text{if } c_2\le x\le 1, & 1\end{array}\right\} =: \tilde{u}_1^*(x)\le u(x)\le \tilde{u}_2^*(x) := \left\{\begin{array}{ll}\frac{1-e^{-\frac{ax}{\nu}}}{1-e^{-\frac{ac_1}{\nu}}} & \text{if } 0\le x\le c_1,\\ \frac{1-e^{-\frac{ax}{\nu}}}{1-e^{-\frac{ac_1}{\nu}}} & \text{if } c_1<x<c_2,\\ 1 & \text{if } c_2\le x\le 1.\end{array}\right.$$

Step 2: We now compute the relative L^2-errors for $\tilde{u}_i^*$, $i=1,2$. Using Equation (15), we have

$$e_1^2 = I_1(1,2)+I_2+I_3 \quad \text{and} \quad e_2^2 = I_1(2,1)+I_3,$$

where the different terms are integrals of the form $\int(\frac{\tilde{u}_i^*}{u}-1)^2dx$,

$$I_1(i,j) := \int_0^{c_i}\left(\frac{1-e^{-\frac{a}{\nu}}}{1-e^{-\frac{ac_j}{\nu}}}-1\right)^2 dx = c_i\left(\frac{1-e^{-\frac{a}{\nu}}}{1-e^{-\frac{ac_j}{\nu}}}-1\right)^2 = O\left(c_i(\frac{\nu}{a})e^{-\frac{2ac_j(\frac{\nu}{a})}{\nu}}\right), \tag{16}$$

$$I_2 := \int_{c_1}^{c_2}\left[\frac{(1-e^{-\frac{ac_1}{\nu}})(1-e^{-\frac{a}{\nu}})}{(1-e^{-\frac{ac_2}{\nu}})(1-e^{-\frac{ax}{\nu}})}-1\right]^2 dx \le \delta\max_{i=1,2}\left(\left[\frac{(1-e^{-\frac{ac_1}{\nu}})(1-e^{-\frac{a}{\nu}})}{(1-e^{-\frac{ac_2}{\nu}})(1-e^{-\frac{ac_i}{\nu}})}-1\right]^2\right)$$
$$= O\left(\delta(\frac{\nu}{a})e^{-\frac{2ac_1(\frac{\nu}{a})}{\nu}}\right), \tag{17}$$

$$I_3 := \int_{c_2}^1\left(\frac{1-e^{-\frac{a}{\nu}}}{1-e^{-\frac{ax}{\nu}}}-1\right)^2 dx = \int_{c_2}^1\left[\sum_{k=1}^{\infty}e^{-\frac{kax}{\nu}}(1-e^{-\frac{a}{\nu}})-e^{-\frac{a}{\nu}}\right]^2 dx = O\left(\frac{\nu}{a}e^{-\frac{2ac_2(\frac{\nu}{a})}{\nu}}\right). \tag{18}$$

As $e^{-\frac{ac_i}{\nu}}<1$ and $e^{-\frac{ax}{\nu}}<1$, we can use geometric series to obtain estimates of the different integrals. Taking only the leading term gives the result for $I_1(i,j)$ and I_3. For I_3, the leading term under the integration is $e^{-\frac{ax}{\nu}}$, because $x\le 1$. For I_2 we also used the monotonicity of the exponential to obtain the bound and then, use once again a geometric series to conclude. In the order notation, we have specified the possible dependence of c_i and δ on the parameter $\frac{\nu}{a}$.

Step 3: We now need to distinguish the different cases in order to complete the proof. Using Equations (16,17,18), we can compute the results shown in Table 1. Finally, we use relation (15) to obtain (14). □

This theorem shows that the approximation quality of the method is similar to the best known coupling methods for this kind of problem, namely the one based on the factorization of the operator, see [10].

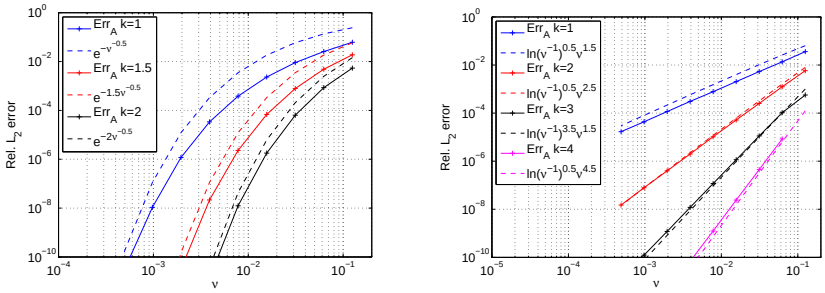
Numerical experiment: We now show a numerical experiment, where we solve (10) with the membership function h_1 as in Equation (13), with

$$h(x) = \delta^{-3}(2x^3-3(c_1+c_2)x^2+6c_1c_2x-c_2^2(3c_1-c_2)),$$

and $h_2 := 1-h_1$. With this decomposition, we solve the advection-diffusion problem if $x\le c_1$, the purely advective model if $x\ge c_2$, and the mixed model in-between. The coupling is done with a spline. We introduce a set of equidistant points $x_i =$

	$c_1 = \text{cst.}$, $\delta = \text{cst.}$	$c_1 = \kappa \left(\frac{\nu}{a}\right)^{1-\varepsilon}$, $\delta = \kappa' \left(\frac{\nu}{a}\right)^{1-\varepsilon}$	$c_1 = \kappa \frac{\nu}{a} \ln(\frac{a}{\nu})$, $\delta = \kappa' \frac{\nu}{a}$	$c_1 = \kappa \frac{\nu}{a}$, $\delta = \kappa' \frac{\nu}{a}$
$I_1(1,2)$	$O(e^{-\frac{2ac_2}{\nu}})$	$O\left(e^{-2(\kappa+\kappa')(\frac{a}{\nu})^{\varepsilon}}\right)$	$O(\ln(\frac{a}{\nu})(\frac{\nu}{a})^{2\kappa+1})$	$O(\frac{\nu}{a})$
$I_1(2,1)$	$O(e^{-\frac{2ac_1}{\nu}})$	$O\left(e^{-2\kappa(\frac{a}{\nu})^{\varepsilon}}\right)$	$O(\ln(\frac{a}{\nu})(\frac{\nu}{a})^{2\kappa+1})$	$O(\frac{\nu}{a})$
I_2	$O(e^{-\frac{2ac_1}{\nu}})$	$O\left(e^{-2\kappa(\frac{a}{\nu})^{\varepsilon}}\right)$	$O((\frac{\nu}{a})^{2\kappa+1})$	$O(\frac{\nu}{a})$
I_3	$O(e^{-\frac{2ac_2}{\nu}})$	$O\left(e^{-2(\kappa+\kappa')(\frac{a}{\nu})^{\varepsilon}}\right)$	$O((\frac{\nu}{a})^{2\kappa+1})$	$O(\frac{\nu}{a})$
e_1^2	$O(e^{-\frac{2ac_1}{\nu}})$	$O\left(e^{-2\kappa(\frac{a}{\nu})^{\varepsilon}}\right)$	$O(\ln(\frac{a}{\nu})(\frac{\nu}{a})^{2\kappa+1})$	$O(\frac{\nu}{a})$
e_2^2	$O(e^{-\frac{2ac_1}{\nu}})$	$O\left(e^{-2\kappa(\frac{a}{\nu})^{\varepsilon}}\right)$	$O(\ln(\frac{a}{\nu})(\frac{\nu}{a})^{2\kappa+1})$	$O(\frac{\nu}{a})$

Table 1 Table of the order of the different integrals I_j.



(a) Case 2: $c_1 = k\frac{\nu}{a}^{1-\varepsilon}$, $\delta = \frac{\nu}{a}^{1-\varepsilon}$, with $a = 1$, $\varepsilon = 0.5$ and $k = 1, 1.5, 2$.

(b) Case 3: $c_1 = k\frac{\nu}{a}\ln(\frac{a}{\nu})$, $\delta = \frac{\nu}{a}$, with $a = 1$ and $k = 1, 2, 3, 4$.

Fig. 1 Results for the cases 2 and 3 of Theorem 1 where we refined the grid keeping $n\nu$ constant. We see that the curves follow the theoretical predictions.

$i \cdot \Delta x$ with $i = 0, \ldots, n+1$ and $\Delta x = 1/(n+1)$. We discretize the problem (10) with an upwind 3-point finite difference scheme. This gives us a system of $2n$ coupled equations. For each component u_j, $j = 1, 2$, we remove from the system all the irrelevant equations, those for which $h_j(x_i) = 0$; this corresponds to the restriction to $\text{Supp}(\Omega_j)$.

In order to illustrate the behavior of the method, we have chosen the cases 2 and 3 in Theorem 1. In both cases, the observed behavior is in very good agreement with the predictions, see Figure 1 where we computed the relative error Err_A between the numerical advection-diffusion solution and its approximation for different parameters. In the two cases shown, the coupling region is moving towards zero when ν is decreasing and we see that the approximation quality depends on how the coupling region is moved, accordingly to Theorem 1. We kept $n\nu$ constant in order to capture the boundary layer that forms when $\nu \to 0$.

4 Conclusion

We presented a new heterogeneous domain decomposition method based on Fuzzy Set Theory. We have shown a concise analysis for a simple, but relevant, model problem which showed that this type of coupling leads to a very efficient heterogeneous domain decomposition method. This method can be viewed as a formalization of a coupling technique for very complex problems, see for example [5, 6] for the coupling between kinetic and hydrodynamic equations. In such a coupling, the partition between the different fuzzy domains can evolve with time and can even adapt automatically to the local conditions using some local criterion, see [6].

We think that such methods have a great potential in various coupling problems and in particular for problems in which the partition into different domains of validity of concurrent approximations is not a priori clear, because they permit to try different criteria by changing only the way the membership functions are defined.

We are currently interested in such a method for the coupling of the diffusion limit of the relativistic Boltzmann equation with a stationary free streaming limit of it. This would be an alternative to the current version of the IDSA, which still has some mathematical issues that need to be fixed, see [2, 3] for more details.

References

1. Achdou, Y., Pironneau, O.: The χ-method for the Navier-Stokes equations. IMA J. Numer. Anal. **13**(4), 537–558 (1993)
2. Berninger, H., Frénod, E., Gander, M.J., Liebendörfer, M., Michaud, J., , Vasset, N.: Derivation of the Isotropic Diffusion Source Approximation (IDSA) for Supernova Neutrino Transport by Asymptotic Expansions SIAM J. Math. Anal., **45**(6), pp. 3229–3430 (2013)
3. Berninger, H., Frénod, E., Gander, M., Liebendörfer, M., Michaud, J., Vasset, N.: A mathematical description of the IDSA for supernova neutrino transport, its discretization and a comparison with a finite volume scheme for Boltzmann's equation. In: Esaim: Proceedings, vol. 38, pp. 163–182 (2012)
4. Brezzi, F., Canuto, C., Russo, A.: A self-adaptive formulation for the Euler/Navier-Stokes coupling. Comput. Methods Appl. Mech. Engrg. **73**(3), 317–330 (1989)
5. Degond, P., Dimarco, G., Mieussens, L.: A moving interface method for dynamic kinetic fluid coupling. J. Comput. Phys. **227**, 1176–1208 (2007)
6. Degond, P., Dimarco, G., Mieussens, L.: A multiscale kinetic-fluid solver with dynamic localization of kinetic effects. J. Comput. Phys. **229**, 4907–4933 (2010)
7. Degond, P., Jin, S.: A smooth transition between kinetic and diffusion equations. SIAM J. Numer. Anal. **42**(6), 2671–2687 (2005)
8. Degond, P., Jin, S., Mieussens, L.: A smooth transition model between kinetic and hydrodynamic equations. J. Comput. Phys. **209**, 665–694 (2005)
9. Discacciati, M., P., G., Quarteroni, A.: Heterogeneous mathematical models in fluid dynamics and associated solution algorithms. Tech. Report MOX 04/2010 (2010)
10. Gander, M.J., Martin, V.: An asymptotic approach to compare coupling mechanisms for different partial differential equations. In: Domain Decomposition Methods in Science and Engineering XX, Lect. Notes Comput. Sci. Eng. Springer-Verlag, 359–366 (2013). In print
11. Liebendörfer, M., Whitehouse, S., Fischer, T.: The isotropic diffusion source approximation for supernova neutrino transport. Astrophys. J. **698**, 1174–1190 (2009)
12. Zadeh, L.: Fuzzy Sets. Information and Control **8**(3), 338–353 (1965)

A New Coarse Grid Correction for RAS/AS

Martin J. Gander[1], Laurence Halpern[2], and Kévin Santugini Repiquet[3]

1 Introduction

It is well known that for elliptic problems, domain decomposition methods need a coarse grid in order to be scalable. One talks about strong scalability of an algorithm, if it permits to solve a problem of fixed size faster in the same proportion that one adds processors. For example if on one processor, a strongly scalable algorithm needs 10 seconds to solve the problem, it would need 1 second using 10 processors. Strong scalability is difficult to achieve already from a theoretical point of view, the limit as the number of processors goes to infinity leads to zero work per processor for a problem of fixed size. One therefore also talks about weak scalability, which means that one can solve a larger and larger problem with more and more processors in a fixed time. For example if a weakly scalable algorithm solves a problem with 100'000 unknowns in 10 seconds using 1 processor, it should be able to solve a problem with 1'000'000 unknowns in the same 10 seconds using 10 processors. Domain decomposition methods with coarse grids attempt to reach this goal.

The most fundamental result for the two level additive Schwarz method is then precisely that the condition number of the preconditioned elliptic problem satisfies the estimate

$$\mathscr{K}(M_{AS}^{-1}A) \leq C(1+\frac{H}{\delta}), \tag{1}$$

where δ denotes the size of the overlap, and H the diameter of the coarse mesh, see the seminal technical report [2], or also the book [11] for a complete and detailed treatment. This result indicates that if one keeps the ratio of the coarse mesh cells to the overlap in a two level additive Schwarz method constant, the method is weakly scalable (as long as the coarse grid solve remains negligible).

Similarly, for substructuring methods, to which the FETI and Balancing Neumann-Neumann methods belong, there is a condition number estimate of the preconditioned system of the form

$$\mathscr{K}(M_{sub}^{-1}A) \leq C(1+\ln(\frac{H}{h}))^2, \tag{2}$$

where now h denotes the mesh size. This theoretical result has been established for the Balancing Neumann-Neumann algorithm in [3, 9], and for the FETI method in [10]; for a complete treatment, see again the book [11]. In overlapping methods,

[1]Université de Genève e-mail: Martin.Gander@unige.ch .[2] Université Paris 13 e-mail: halpern@math.univ-paris13.fr .[3] Institut Mathématiques de Bordeaux, CNRS UMR5251, MC2, INRIA Bordeaux - Sud-Ouest e-mail: Kevin.Santugini@math.u-bordeaux1.fr

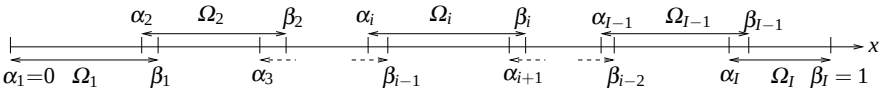


Fig. 1 Decomposition into many subdomains for the one dimensional model problem

the mesh size h is often related to the overlap parameter δ, since the overlap is in general just one or a few mesh cells, and this permits us to compare (1) and (2).

It is also very easily possible to understand intuitively why such a coarse level correction is necessary, if one wants to obtain a scalable method. For the simple model problem,

$$(\eta - \partial_{xx})u = 0, \quad u(0) \text{ and } u(1) \text{ given}, \tag{3}$$

we consider the parallel Schwarz method introduced by Lions [8] for the decomposition shown in Figure 1,

$$\begin{aligned} (\eta - \partial_{xx})u_i^n &= 0 \quad \text{in } \Omega_i, \\ u_i^n(\alpha_i) = u_{i-1}^{n-1}(\alpha_i), &\quad u_i^n(\beta_i) = u_{i+1}^{n-1}(\beta_i), \end{aligned} \tag{4}$$

which is a one level method, and is equivalent to RAS (restricted additive Schwarz [1]), see [4] and [5] for a proof of equivalence. We show in Figure 2 the first few iterations of algorithm (4): in the top row, for the case of two subdomains, we clearly see that both iterates on the left and right subdomain start to converge with the first iterations toward the solution, which is a straight line in this example with $\eta = 0$, whereas with sixteen subdomains in the bottom row, the subdomains on the left remain at zero, since communication in this algorithm is only local between the subdomains.

2 Geometric Investigation of the Coarse Grid Correction

In order to obtain a scalable algorithm, one can introduce a second level solve like in multigrid: one simply introduces for the fine discretization $Au = f$ of (3) a coarse

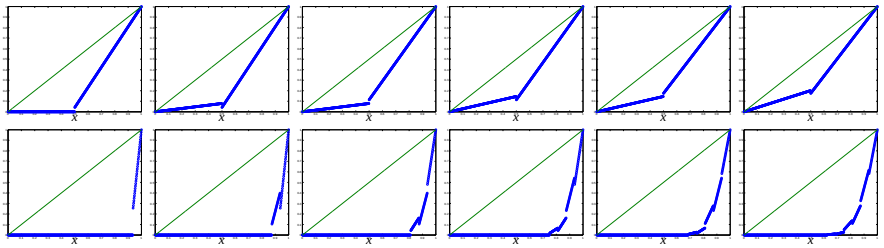


Fig. 2 First iterations of Lions parallel Schwarz methods (equivalent to RAS) for two subdomains in the top row, and sixteen subdomains in the bottom row, $\eta = 0$

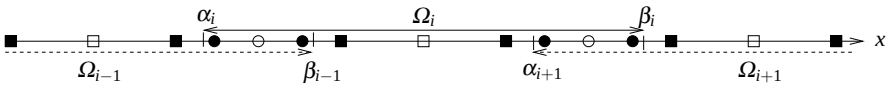


Fig. 3 Various choices to place coarse grid nodes: center of subdomains (empty squares), center of overlaps (empty circles), in the overlap to the left and right of the RAS discontinuity (filled circles) and an equal number of coarse grid points within the subdomains for a fair comparison (filled squares)

grid, and then, after each iteration of algorithm (4), performs the correction

$$\begin{aligned} r_n &= f - Au_n; \\ r_c &= Rr_n; \\ u_c &= A_c^{-1} r_c; \\ u_n &= u_n + Eu_c; \end{aligned} \tag{5}$$

using standard components. In our example, we use for the extension E linear interpolation, for the restriction R the extension transposed and normalized, and for the coarse matrix the Galerkin projection $A_c = RAE$. A classical choice for the coarse grid is to put one grid point into the center of each subdomain as shown by the empty square in Figure 3. This leads for our example to the convergence result shown in Figure 7 on the left. We clearly see that without coarse grid, the convergence slows down as we add subdomains, whereas with the coarse grid, the convergence curves remain the same, the algorithm is scalable.

In order to see geometrically how the coarse grid correction (5) works, we now visualize in each iteration step how it operates: we show in Figure 4 for the case of four subdomains the iterates before the coarse grid correction, then the residual, the best coarse correction possible and the one actually computed, and finally the iterates after the coarse grid correction. We clearly see that the coarse grid correction is effective: after one coarse grid correction, in the top row, the approximate solution is already very close for all subdomains to the solution sought. We see however also a very unnatural kink appearing in the corrected approximation on the right. Looking at the middle picture of the top row, we see that the residual is concentrated in the center of the overlaps. This is because in RAS, subdomain solutions are composed piecewise, and subdomain solutions satisfy the equations in the subdomains (one says they are harmonic), and thus have zero residual there. The coarse correction computed with grid points in the center of each subdomain are not suitable to correct such a residual support well, as one can see in the middle figure in each row: the residual is smeared out into the subdomains, instead of being corrected in the overlap.

This indicates that coarse grid degrees of freedom in our example should be placed in the overlap, in order to avoid the smearing of the residual into the subdomains, and ideally one should have one degree of freedom on each side of the non-zero residual location, in order to capture the 'jump' in the ideal correction shown in the middle column, see the filled circles in Figure 3. The best coarse space must have as a range such types of corrections. We show in Figure 5 for the same

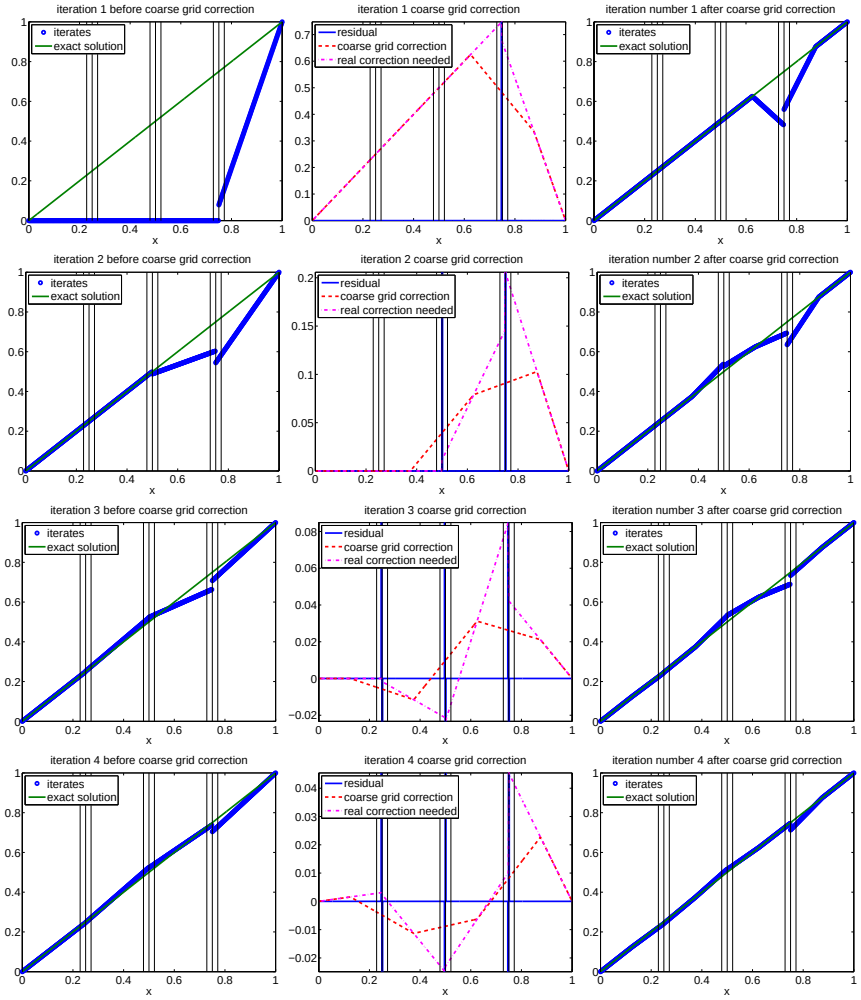


Fig. 4 On each line the iterates before the coarse correction (left), residual, best possible coarse correction and coarse correction actually computed (middle), and iterates after the coarse correction (right) for the first few iterations of the Lions parallel Schwarz method with coarse correction

example what happens with this new coarse grid correction. The result is striking: we obtain convergence of the Schwarz algorithm with this coarse grid correction in two iterations, independently of the number of subdomains. Under the conditions

(i) The coarse grid nodes are in the overlap and can capture the discontinuity from RAS,
(ii) The coarse grid functions satisfy the homogeneous equation,

one obtains a direct solver! In order to illustrate that it is important for the coarse grid shape functions to be harmonic, we show in Figure 6 what happens when we

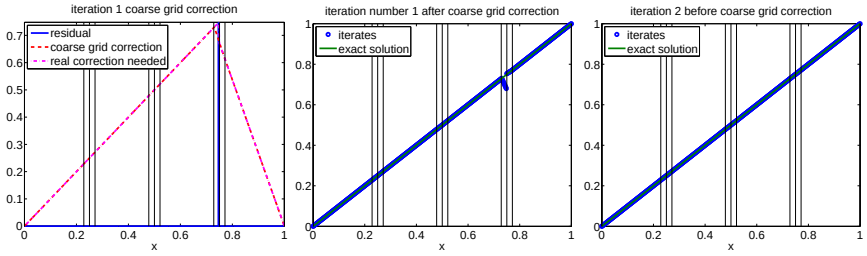


Fig. 5 Residual, best possible coarse grid correction and coarse correction actually computed with good placement of coarse grid nodes (left), iterate after the new coarse grid correction (middle), and iterate after the Schwarz correction (right) starting with the same initial configuration as shown on the top left in Figure 4

solve a problem with $\eta = 10$, and still use piecewise linear coarse shape functions. We clearly do not obtain the solution any more after two iterations, but still a very rapidly converging method, note the different scaling in the residual plot on the left of Figure 6! In order to finally compare with a classical two level additive Schwarz method (AS), and measure the influence of using a Krylov method to accelerate the iteration, we present in Figure 7 on the right the convergence histories for this example. It is well known that AS does not converge without Krylov acceleration, which explains the plateau observed in Figure 7. But even with Krylov acceleration, the method is much slower than RAS with the new coarse grid placement. We also notice that RAS now does basically not need Krylov acceleration any more, convergence with and without Krylov is very similar.

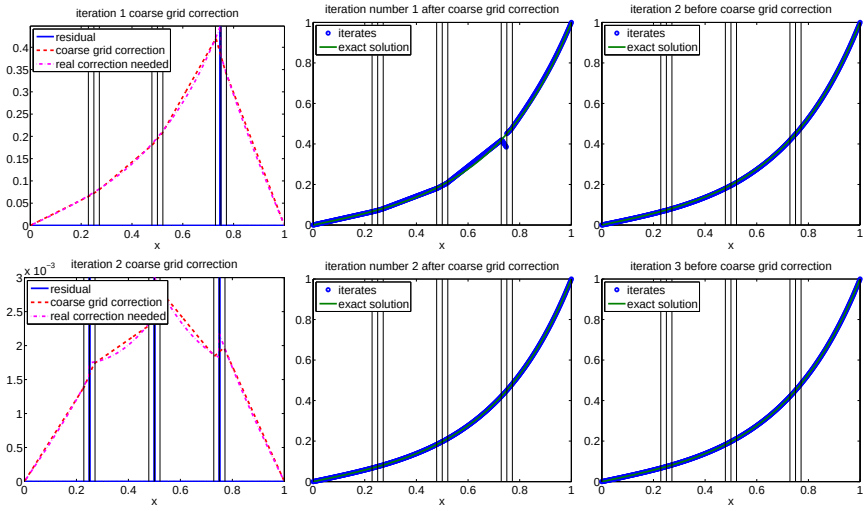


Fig. 6 Example with $\eta = 10$, but otherwise the same configuration as in Figure 5

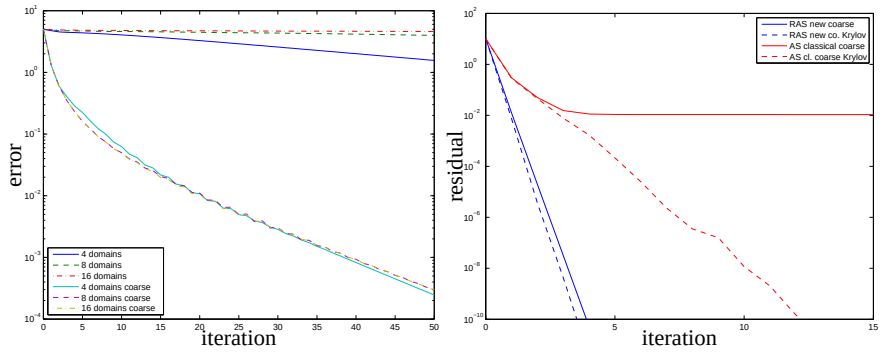


Fig. 7 Iteration versus error for Lions parallel Schwarz algorithm with and without coarse grid on the left, and comparison of AS with classical coarse grid and RAS with optimally placed coarse grid with and without Krylov acceleration on the right

The key question is: can we learn anything from this simple one dimensional example for a problem in higher dimensions? According to the design rule 1. above, the coarse grid needs to have nodes in the overlap, and enough to capture an arbitrary residual located there, as shown in Figure 8 on the left. Then one can prove that we still get a direct solver, provided design rule 2. above is also satisfied. It is interesting at this point to indicate a relation of this coarse grid correction and the optimal transmission operator introduced in [6], which leads to convergence of an optimized Schwarz method in two iterations, independently of the number of subdomains and subdomain configuration, even with crosspoints! The transmission operator also contains a coarse grid component there, and it needs precisely the same traces as our presently proposed coarse grid, and one can find a complete proof at the algebraic level on convergence in two iterations in [6]. Similarly, for a banded matrix, there is also an optimal transmission operator in [7], which again involves

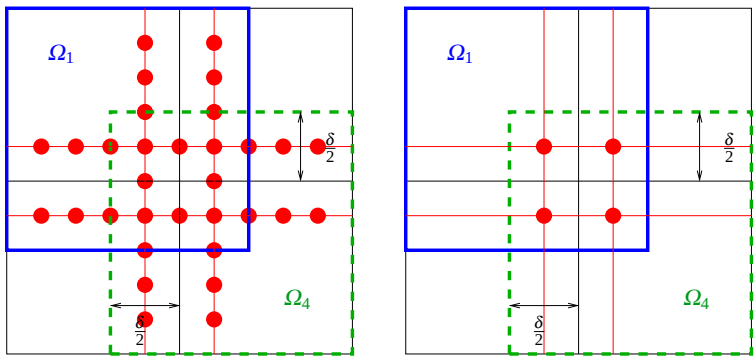


Fig. 8 Optimal coarse grid in two dimensions, and a simple approximation, extending the 1d optimal placement in tensor form

the same global traces. Naturally, these methods are related, but the precise relation is non-trivial and will be developed elsewhere.

The coarse space indicated in Figure 8 on the left is however very expensive, it requires many degrees of freedom, and also a solve for each in order to obtain harmonic shape functions. A much cheaper alternative is indicated in Figure 8 on the right: one simply places four coarse grid nodes around the cross point of the decomposition. One can then again use Q1 coarse shape element functions, which are harmonic. We show in Figure 9 the convergence histories we obtain for the Laplace equation on the unit square, decomposed into 16×16 subdomains, using 256×256 gridpoints. On the left we used the Lions Schwarz method with a coarse grid (equivalent to two level RAS) with overlap $3h$. We show the result for the

- classical placement of one coarse grid node in the center of each subdomain (classical Q1, empty square in the 1d Figure 3),
- one node at each crosspoint (Q1 in overlap, empty circle in the 1d Figure 3), in order to illustrate that really one node is not enough for the jumps in RAS,
- four nodes per subdomain equally spaced (Q1 fair, filled square in the 1d Figure 3) with the same number of coarse grid points as the optimized coarse grid for a fair comparison, and
- four nodes around the crosspoints (optimized Q1, filled circle in the 1d Figure 3), with the same number of coarse grid points as Q1 fair.

Clearly the optimized placement of the coarse grid nodes leads to a substantially faster method than all the other choices.

In Figure 9 on the right we show the corresponding result for AS with minimal overlap h. It is interesting to note that for minimal overlap, the influence of the placement of the coarse nodes is even more important, and one obtains a much faster method than with any of the other coarse grid node placements in this two dimensional example.

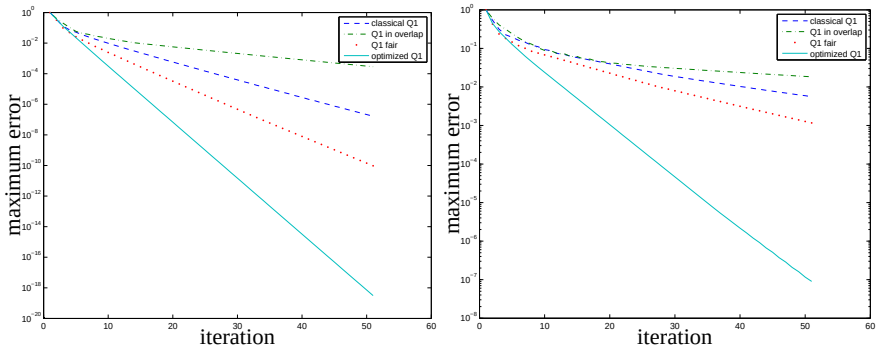


Fig. 9 Convergence histories for two level RAS with various coarse grid node placements on the left and overlap $3h$, and on the right for AS (additive Schwarz) with overlap h

3 Conclusions

We explained geometrically the interplay between Schwarz iterations and coarse grid corrections. Our example in one dimension revealed that in addition to having harmonic coarse space shape functions, it is also very important where the coarse grid nodes are placed. Optimal placement in one dimension is in the overlap, which leads to a method that converges in two iterations, independently of the number of subdomains. In higher spatial dimensions, it is still possible to construct such a coarse grid correction, but one has to use a number of degrees of freedom proportional to the skeleton of the decomposition. Using however a simple approximation, placing only few degrees of freedom around the crosspoints, leads already to a much faster iterative method than placing coarse nodes as it is done traditionally somewhere within the subdomains. Several theoretical results are already available, though in the different context of transmission conditions, see [6] and [7], and we are currently working on a rigorous error analysis of this new idea. It is also an open question how such an optimized coarse grid would have to look like for a general decomposition of a general domain, our examples here having been simple squares.

References

1. Xiao-Chuan Cai and Marcus Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21:239–247, 1999.
2. Maksymilian Dryja and Olof B. Widlund. An additive variant of the Schwarz alternating method for the case of many subregions. Technical Report 339, also Ultracomputer Note 131, Department of Computer Science, Courant Institute, 1987.
3. Maksymilian Dryja and Olof B. Widlund. Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. *Comm. Pure Appl. Math.*, 48(2):121–155, February 1995.
4. Evridiki Efstathiou and Martin J. Gander. Why Restricted Additive Schwarz converges faster than Additive Schwarz. *BIT Numerical Mathematics*, 43(5):945–959, 2003.
5. Martin J. Gander. Schwarz methods over the course of time. *ETNA*, 31:228–255, 2008.
6. Martin J. Gander and Felix Kwok. Optimal interface conditions for an arbitrary decomposition into subdomains. In *Domain Decomposition Methods in Science and Engineering XIX*. Springer LNCSE, 2010.
7. Martin J. Gander, Sebastien Loisel, and Daniel Szyld. An optimal block iterative method and preconditioner for banded matrices with applications to PDEs on irregular domains. *SIAM Journal on Matrix Analysis and Applications*, 33(2):653–680, 2012.
8. Pierre-Louis Lions. On the Schwarz alternating method. I. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42, Philadelphia, PA, 1988. SIAM.
9. Jan Mandel and Marian Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comp.*, 65:1387–1401, 1996.
10. Jan Mandel and Radek Tezaur. Convergence of a substructuring method with Lagrange multipliers. *Numer. Math.*, 73:473–487, 1996.
11. Andrea Toselli and Olof Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, 2005.

Aggregation-based aggressive coarsening with polynomial smoothing

James Brannick[1]

Abstract This paper develops an algebraic multigrid preconditioner for the graph Laplacian. The proposed approach uses aggressive coarsening based on the aggregation framework in the setup phase and a polynomial smoother with sufficiently large degree within a (nonlinear) Algebraic Multilevel Iteration as a preconditioner to the flexible Conjugate Gradient iteration in the solve phase. We show that by combining these techniques it is possible to design a simple and scalable algorithm. Results of the algorithm applied to graph Laplacian systems arising from the standard linear finite element discretization of the scalar Poisson problem are reported.

1 Introduction

This paper concerns the development of an algebraic multigrid (AMG) method for solving the (graph) Laplacian problem. The corresponding linear system is defined in terms of the following bilinear form:

$$(Au,v) = \sum_{e\in\mathscr{E}} w_e\delta_e u\delta_e v + \sum_{i\in S_b} d_i u_i v_i = (f,v), \tag{1}$$

where $\mathscr{G} = (\mathscr{V},\mathscr{E})$ denotes an unweighted connected graph, $\mathscr{V}$ and $\mathscr{E}$ denote the set of vertices and edges of $\mathscr{G}$, respectively, and $\delta_e u = (u_i - u_j)$ for $e = (i,j) \in \mathscr{E}$. Note that the *lower-order terms*, $d_i u_i v_i, i \in S_b$, are included to account for various types of *boundary conditions* for problems originating from discretization of partial differential equations (PDEs). If the lower-order terms are omitted and the wieghts $w_e = 1$, then the variational problem reduces to the graph Laplacian for a graph $\mathscr{G}$ that we focus on here. The graph Laplacian, A, is then a symmetric and positive semi-definite matrix and its kernel is the space spanned by the constant vector.

The main aim of the paper is to study the use of polynomial smoothing together with aggressive unsmoothed aggregation-based algebraic multigrid (UA-AMG) coarsening in developing an AMLI-cycle or k-cycle preconditioner [2] for the graph Laplacian system. We consider the recently proposed polynomial based on the best approximation to x^{-1} in the uniform norm [10] in formulating the proposed UA-AMG algorithm. A multilevel smoothed aggregation (SA) AMG algorithm using polynomial smoothers based on Chebychev approximations and its V-cycle convergence analysis are found in [13]. We note that, these results are also used in [10]

[1] Department of Mathematics, The Pennsylvania State University, University Park, PA 16802, USA
e-mail: brannick@psu.edu

to derive an SA two-level preconditioner with polynomial smoothing for diffusion problems. In both methods, the polynomial approximation is used to form (1) a smoother for the interpolation operator and (2) a relaxation scheme for the solver. These preconditioners yield uniformly convergent methods provided polynomials of sufficiently large degree are used in both steps. Further development and analysis of polynomial smoothers are found in [1] and [8, 3].

Here, we consider an approach in which the polynomial smoother is used as the relaxation scheme in the AMG solver and interpolation is based on UA-AMG framework. We show that using such plain aggregation based aggressive coarsening with a polynomial smoother in a AMLI cycle or k-cycle leads to a uniformly convergent method. Generally, the use of unsmoothed (or plain) aggregation to construct the coarse space without the use of interpolation smoothing has been observed to result in slow convergence of a V-cycle multilevel iterative solver. We note that recently it has been shown that plain aggregation-based coarsening approaches can lead to effective solvers for a variety of problems provided AMLI or k-cycles are used, e.g, such approaches have been developed and analyzed for the graph Laplacian in [11], for more general M matrices in [12, 7], and for problems in quantum dynamics in [4]. Generally, the use of AMLI cycles and UA-AMG typically leads to low grid and operator complexities, limited fill-in in the coarse level operators, and reduces the arithmetic complexity in the setup phase substantially. The gains in the solve phase are often less pronounced since AMLI- and NAMLI-cycles use additional coarse-level corrections to accelerate convergence of the UA-AMG method.

In Section 2, we introduce a graph partitioning algorithm for constructing the coarse space. Then, in Section 3, we establish an approximation property for such piecewise constant coarse spaces, which together with the stability estimates for such methods found in [7], gives a spectral equivalence result that holds for the corresponding two-level method applied to graph Laplacian on general graphs. The resulting estimate depends on the degree of the polynomial smoother and the coarsening ratio, i.e., the cardinality of the aggregates, and thus provides a way to adjust the polynomial degree to compensate for aggressive coarsening. We note that the result is a special case of the general result found in [10]. In the last section, we provide numerical experiments of the proposed multigrid approach applied to the graph Laplacian and show that the coarsening can be quite aggressive and still only a low degree polynomial is needed to obtain a scalable AMLI or k-cycle preconditioner.

2 Subspaces by graph partitioning and graph matching

We define a graph partitioning of $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ as a set of connected subgraphs $\mathscr{G}_i = (\mathscr{V}_i, \mathscr{E}_i)$ such that $\cup_i \mathscr{V}_i = \mathscr{V}, \quad \mathscr{V}_i \cap \mathscr{V}_j = \emptyset, \quad i \neq j$. In this paper, all subgraphs are assumed to be non empty and connected. The simplest non trivial example of such a graph partitioning is a matching, i.e, a collection (subset $\mathscr{M}$) of edges in $\mathscr{E}$ such that no two edges in $\mathscr{M}$ are incident. For a given graph partitioning, subspaces of $V = \mathbb{R}^{|\mathscr{V}|}$ are defined as

$$V_H = \{v \in V |\ v = \text{ constant on each } \mathscr{V}_i\ \}. \tag{2}$$

Note that each vertex in $\mathscr{G}$ corresponds to a connected subgraph $\mathscr{G}_i$ of $\mathscr{G}$ and every vertex of $\mathscr{G}$ belongs to exactly one such component. The vectors from V_H are constants on these connected subgraphs. The ℓ_2 orthogonal projection on V_H, which is denoted by Q, is defined as follows:

$$(Qv)_i = \frac{1}{|\mathscr{V}_k|} \sum_{j \in \mathscr{V}_k} v_j, \quad \forall i \in \mathscr{V}_k. \tag{3}$$

Given a graph partitioning, the coarse graph $\mathscr{G}_H = \{\mathscr{V}_H, \mathscr{E}_H\}$ is defined by assuming that all vertices in a subgraph form an equivalence class and that $\mathscr{V}_H$ and $\mathscr{E}_H$ are the quotient set of $\mathscr{V}$ and $\mathscr{E}$ under this equivalence relation. That is, any vertex in $\mathscr{V}_H$ corresponds to a subgraph in the partitioning, and the edge (i, j) exists in $\mathscr{E}_H$ if and only if the i-th and j-th subgraphs are connected in the graph $\mathscr{G}$.

The algorithm we use in forming a graph partitioning is a variant of the approach we developed and tested for graphics processing units in [5]. The procedure iteratively applies the following two steps:

(A) Construct a set S which contains coarse vertices by applying a maximal independent set algorithm to the graph of A^k.
(B) Construct a subgraph for each vertex in S by collecting vertices and edges of the neighbors of vertices in S.

3 Two-level preconditioner with polynomial smoothing for the graph Laplacian

A variational two-level method with one post smoothing step is defined as follows. Given an approximation $w \in V$ to the solution u of the graph Laplacian system, an update $v \in V$ is computed in two steps

(i) $y = w + PA_H^{\dagger}P^T(f - Aw), \quad A_H = P^TAP.$
(ii) $v = y + R(f - Ay).$

We use $\dagger$ to denote the pseudo inverse of a matrix. The corresponding error propagation operator of the two-level method is given by

$$E_{TL} = (I - RA)(I - \pi_A), \quad \pi_A = PA_H^{\dagger}P^TA.$$

Here, E_{TL} is nonsymmetric and, thus, we consider the following *symmetrization* to form the two-level preconditioner: $B = (I - E_{TL}E_{TL}^*)A^{\dagger}$, with * denoting the adjoint with respect to the energy inner product $(\cdot,\cdot)_A$. We note that $|E_{TL}|_A^2 = \rho(I - BA)$, where $\rho(X)$ is the spectral radius of the matrix X. Further, if $\bar{R}$ satisfies $(I - \bar{R}A) = (I - RA)^2$ so that $\bar{R} = 2R - RAR$, then using that π_A is an A-orthogonal projection on range(P), it follows by direct computation that $B = \bar{R} + (I - RA)\,PA_H^{\dagger}P^T\,(I - AR)$.

In [10], a spectral equivalence result for the preconditioner B using a polynomial smoother based on the best approximation to x^{-1} on a finite interval $[\lambda_0, \lambda_1]$, $0 < \lambda_0 < \lambda_1$, in the uniform norm ($\|\cdot\|_\infty$) is derived. Here, $\lambda_0 > 0$ is any lower bound on the spectrum of A and $\lambda_1 = \|A\|_{\ell_\infty}$ is an approximation to $\rho(A)$. The unique solution to the minimization problem

$$q_m(x) = \arg\min\{\|\frac{1}{x} - p\|_{\infty,[\lambda_0,\lambda_1]}, \quad p \in \mathscr{P}_m\}, \tag{4}$$

determines the polynomial approximation of degree m. For details on the three-term recurrence used in its construction we refer to [10]. Define

$$E_m := \max_{x\in[\lambda_0,\lambda_1]} |1 - xq_m(x)| = \max_{x\in[\lambda_0,\lambda_1]} x \cdot \left|\frac{1}{x} - q_m(x)\right|.$$

Then, since λ_1 is a point of Chebyshev alternance from [10, Theorem 2.1 and Equation (2.2)] for the error of approximation E_m we have

$$E_m = \lambda_1 \left|\frac{1}{\lambda_1} - q_m(\lambda_1)\right| = \left[\frac{2\lambda_1}{\lambda_1 - \lambda_0}\right] \cdot \left[\frac{\delta^m}{a^2 - 1}\right] = \frac{2\kappa\delta^m}{(\kappa-1)(a^2-1)}.$$

Here, we have denoted $\kappa = \frac{\lambda_1}{\lambda_0}$, $\delta = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$, and $a = \frac{\kappa+1}{\kappa-1}$. Computing the error E_m then gives

$$E_m = \frac{\delta^m(\kappa-1)}{2}.$$

A restriction on the degree m is given by the requirement that $q_m(\lambda_1) > 0$. A sufficient condition for the positivity of this polynomial (and also necessary condition in many cases) is that $\frac{1}{\lambda_1} - E_m > 0$. Thus, we need to find the smallest m such that both $E_m < \rho$ and $q_m(\lambda_1) > 0$. We then have that the polynomial is positive if

$$\frac{\delta^m(\kappa-1)}{2} \le \frac{1}{\lambda_1} \quad \Rightarrow \quad \delta^m \le \frac{2}{\lambda_1(\kappa-1)}.$$

We note that from this it follows that $R = q_m(A)$ and hence $\bar{R}$ are symmetric and positive definite, implying that the smoother in convergent in A-norm. Also, to guarantee a damping factor less than ρ on the interval $[\lambda_0, \lambda_1]$, we have

$$\frac{\delta^m(\kappa-1)}{2} \le \rho \quad \Rightarrow \quad \delta^m \le \frac{2\rho}{\kappa-1}.$$

Thus, the minimal m that guarantees both properties are satisfied is given by

$$m \ge \frac{1}{|\log\delta|} \max\left\{\left|\log\frac{2\rho}{\kappa-1}\right|, \left|\log\frac{2}{\lambda_1(\kappa-1)}\right|\right\}. \tag{5}$$

The spectral equivalence result that we adopt to analyze a two-level method based on plain aggregation with this polynomial smoother follows from this smoothing es-

timate and the assumptions of stability and an approximation property of the coarse space V_H: for any $v \in V_h$,

$$c_p^{-1}\|v - Qv\|^2 + |v - Qv|_A^2 \leq c_1 |v|_A^2, \tag{6}$$

where $|\cdot|_A$ denotes the A semi norm. Recall that in this paper Q is the ℓ_2 projection on the span of $\{1_l\}_{l=1}^{n_H}$ (p.w. constant projection) and, thus, this inequality holds also in the case that v is in the kernel of A, since then, all the terms are equal to zero.

Assume that V_H is such that the above approximation and stability assumptions hold and the polynomial q_m is chosen such that (5) holds for a fixed value λ_0. Then, the following spectral equivalence holds

$$v^T A v \leq v^T B^\dagger v \leq K_{TG}\, v^T A v, \quad K_{TG} = 8 + 8c_1 \left[c_{n_z} c_p c_s + 1\right]. \tag{7}$$

This result is a special case of Theorem 4.6 in [10], refined for unsmoothed aggregation applied to the graph Laplacian. Here, c_{n_z} is a constant that depends on the number of nonzeros per row of A, the constant c_1 involves the stability of Q in A-norm and the constant c_p arises from the weak approximation property, and as we show below, depends on the cardinality and the diameter of the subgraphs in the graph partitioning. The constant $c_s = \frac{\ln^2 m}{m^2}$, where m is the degree of the polynomial. Thus, given a partitioning of the fine-level graph into subgraphs, $\mathscr{G} = \cup_{l=1}^{n_H} \mathscr{G}_l$, it is possible to choose the degree of the polynomial m sufficiently large to control the constant c_p and hence K_{TG} in the above spectral equivalence estimate. This result is derived from the following estimate (see Corollary 4.4 in [10])

$$v^T B^\dagger v \leq 4 \inf_{v_h \in V_H} \left[|v_H|_A^2 + \lambda c_s \|v - v_H\|^2 + |v - v_H|_A^2 \right]. \tag{8}$$

A similar result for smoothed aggregation based on Chebyshev polynomial approximations is found in [8].

Next, we establish the approximation property for the p.w. constant coarse space V_H as defined in (2) for the graph Laplacian. Suppose that $\mathscr{V} = \{1, \ldots, n\}$ is partitioned into nonoverlapping subsets: $\mathscr{V} = \cup_{l=1}^{n_H} \mathscr{V}_l, n_l = |\mathscr{V}_l|$. Each set of vertices defines a subgraph $\mathscr{G}_\ell$ whose vertex set is $\mathscr{V}_l$ and whose edges $\mathscr{E}_l$ are a subset of $\mathscr{E}$, where $(i, j) \in \mathscr{E}_l$ if and only if *both* i and j are in $\mathscr{V}_l$. Denote the graph Laplacian associated with the subgraph $\mathscr{G}_l$ by A_l. Let 1 denote the constant vector on $\mathscr{V}$ and 1_l the constant vector on $\mathscr{V}_l$ extended by 0 outside $\mathscr{V}_l$. Let λ_l be the smallest positive eigenvalue of the graph Laplacian on $\mathscr{G}_l$, namely, λ_l is defined as $\lambda_l = \min_{v:\,(v,1_l)=0} \dfrac{(A_l v, v)}{\|v\|^2}$. Here, the minimum is taken over all $v \in \mathbb{R}^{n_l}$. Given $v \in \mathbb{R}^n$ define $\|v\|_{\mathscr{G}_l}^2 = \sum_{j \in \mathscr{V}^l} v_j^2$, which is the ℓ_2 norm on the subgraph $\mathscr{G}_l$. Now, since $((v - Qv), 1_l) = 0$, we have $\|v - Qv\|_{\mathscr{G}_l}^2 \leq \lambda_l^{-1} \sum_{e \in \mathscr{E}_l} (\delta_e v)^2$. Thus,

$$\|v - Qv\|^2 = \sum_{l=1}^{n_c} \|v - Qv\|_{\mathscr{G}_l}^2 \leq \sum_{l=1}^{n_c} \lambda_l^{-1} \sum_{e \in \mathscr{E}_l} (\delta_e v)^2 \leq c_p \sum_{e \in \mathscr{E}} (\delta_e v)^2 = c_p (Av, v). \tag{9}$$

The last step follows from the definition of c_p and the observation that since $\cup_l \mathscr{E}_\ell \subset \mathscr{E}$,we have that for any $v \in \mathbb{R}^n$, $\sum_{l=1}^{n_H} \sum_{e \in \mathscr{E}_l} (\delta_e v)^2 \leq \sum_{e \in \mathscr{E}} (\delta_e v)^2 = (Av, v)$. Note that this latter result holds since the second sum is over a larger set. For *shape regular* subgraphs, $\mathscr{G}_l$, the local constants λ_l^{-1} can be bounded in terms of $|\mathscr{V}_l| \cdot \mathrm{diam}(\mathscr{G}_l)$ using Cheeger's inequality [9]. Here, $\mathrm{diam}(\mathscr{G}_l)$ denotes the diameter of the longest path in the lth subgraph. A similar technique is considered in [12], in which the constants λ_ℓ^{-1} are computed by solving local eigenvalue problems.

In [6], commuting relations involving a certain projection, Π, the p.w. constant projection Q, the discrete gradient operator, B, and B^T on the graph, $\mathscr{G}$, are introduced and are then used to derive a stability estimate of the form

$$|Q|_A^2 = \sup_{v:\,(v,1)=0} \frac{|Qv|_A^2}{|v|_A^2} \leq \|\Pi\|^2 \leq c_0 \,,$$

where c_0 is a constant that depends on the shape and alignment of the subgraphs, but not on the dimension, $|\mathscr{V}|$, of the graph Laplacian, A. It is noteworthy that this bound holds for general graphs with few assumptions and, further, that, since Π is constructed one row at a time, this estimate allows local energy estimates that can be used in forming the graph partitioning. A similar approach was considered in [11].

Given the above approximation and stability estimates and using that $|v_H|_A \leq c_0 |v|_A$, $v_H = Qv$, it follows that the inequality in (6) holds with $c_1 = 2c_0 + 3$ and c_p given in (9). This, in turn, implies the spectral equivalence of the two-level preconditioner based on a p.w. constant coarse space V_H for the graph Laplacian. We remark that the Galerkin coarse-level operator $A_H = P^T A P$ is generally a weighted graph Laplacian of the form $A_H = B_H^T D B_H$, where D is a diagonal weight matrix with strictly positive entries and B_H is the discrete gradient operator defined on the coarse graph $\mathscr{G}_H(\mathscr{V}_H, \mathscr{E}_H)$. Similar stability and approximation properties of piecewise constant coarse spaces can be established in this more general setting as well and, then, a similar proof of the spectral equivalence result follows with minor modifications. Alternatively, it is possible to replace the weighted graph Laplacian with an unweighted one on the same graph and derive a spectral equivalence result between the two. The latter result, in turn, again can be used to establish a spectral equivalence result for this modified two-level method.

4 Numerical results

We apply the proposed aggregation based preconditioner to graph Laplacians resulting from finite element discretizations of the scalar Laplace problem. We consider both stationary AMLI-cycle and N-AMLI-cycle (k-cycle) preconditioners. For details on the theory and the implementation of the AMLI and N-AMLI methods we refer to [2]. In the AMLI approach, we use the polynomial based on the best approximation to x^{-1} in the uniform norm to form a the preconditioner between any two successive levels of the multilevel hierarchy, see [10]. In the N-AMLI-cycle,

a nonlinear PCG (NPCG) method is applied recursively to solve the coarse-level equations. The AMLI-cycle is used as a preconditioner for the CG method on the finest level and the N-AMLI-cycle is applied as a preconditioner to the NPCG iteration. To limit the memory requirements of the nonlinear scheme we restart the outer fine-level NPCG method every five iterations.

In all tests, the maximal independent set algorithm used in the aggregation process for constructing the coarse spaces is applied to the graph of A^4, yielding a coarsening factor of roughly $n/n_H = 30$ between any two successive levels. The problem is coarsened until the size of the coarsest level is less than 100. As the relaxation method in the multilevel solver we use the polynomial smoother based on the best approximation to x^{-1} on the interval $[\lambda_0, \lambda_1]$, where the estimate of the largest eigenvalue is computed as $\lambda_1 = \|A\|_{\ell_\infty}$ and we set $\lambda_0 = \lambda_1/10$. Thus, taking the degree as $m = 4$ in the polynomial smoother ensures the inequality (5) holds. We test W-cycle AMLI and N-AMLI preconditioners with such smoother. The stopping criteria for the flexible preconditioned conjugate gradient iteration is set to a 10^{-8} reduction in the relative A norm of the error and the number of iterations needed to reach this tolerance in the different tests are reported.

In Table 4, we report results of the proposed method for graph Laplacians arising from discretizing the Poisson problem on structured and unstructured meshes. We compare the performance of a stationary AMLI with a N-AMLI, both using the same multilevel hierarchy obtained by applying the aggregation algorithm to the same Poisson problem with Neumann boundary conditions discretized using standard linear Finite Elements. For the structured meshes we consider a 2d unit square domain with n^2 unknowns (left) and a 3d unit cube domain with n^3 unknowns (middle). Results for more general graphs (right), coming from unstructured meshes resulting from triangulations of the 3d unit cube, are also included. The unstructured mesh is formed by adding a random vector of length $h/2$, where h is the grid length, to each vertex of a structured triangulation, followed by a Delaunay triangulation. The (AMLI) N-AMLI method yields a (nearly) scalable solver with low grid and operator complexities – in all tests the grid complexities $\frac{\sum_{j=0}^{J} n_j}{n_0}$ were less than 1.03 and the operator complexities $\frac{\sum_{j=0}^{J} nnz(A_j)}{nnz(A_0)}$ were less than 1.04.

2d struct.		
n	AMLI	N-AMLI
512^2	20	19
1024^2	22	20
2048^2	23	21
4096^2	24	21

3d struct.		
n	AMLI	N-AMLI
32^3	22	20
64^3	23	22
128^3	23	22
256^3	25	22

3d unstruct.		
n	AMLI	N-AMLI
32^3	24	21
64^3	25	23
128^3	27	24
256^3	28	24

Table 1 Results of $W(1,1)$ AMLI and nonlinear AMLI preconditioners with degree $m = 4$ polynomial smoother for the Poisson problem.

5 Conclusion

An algebraic graph partitioning algorithm for aggressive coarsening is developed and a two-level convergence theory of the resulting solver with polynomial smoother is developed. It is shown numerically that the resulting N-AMLI approach with polynomial smoother yields an efficient solver for graph Laplacian problems coming from Finite Element discretizations of the Poisson problem. The graph partitioning algorithm, intended for unweighted graphs, is designed to select shape regular aggregates of arbitrary size and, thus, can be used to obtain predefined coarsening factors. The use of an unsmoothed aggregation form of aggressive coarsening results in low overall grid and operator complexities and limited fill-in in the coarse-level operators. It further significantly simplifies the triple matrix product to simple summations of entries of the fine-level matrix.

References

1. Mark Adams, Marian Brezina, Jonathan Hu, and Ray Tuminaro. Parallel multigrid smoothing: polynomial versus gauss-seidel. *J. Comp. Phys*, 188:593–610, 2003.
2. O. Axelsson and P. S. Vassilevski. Algebraic multilevel preconditioning methods. I. *Numer. Math.*, 56(2-3):157–177, 1989.
3. Allison H. Baker, Robert D. Falgout, Tzanio V. Kolev, and Ulrike Meier Yang. Multigrid smoothers for ultraparallel computing. *SIAM Journal on Scientific Computing*, 33(5):2864–2887, 2011.
4. J. Brannick, R. Brower, M Clark, J Osborne, and C. Rebbi. Adaptive multigrid for lattice QCD. *Phys. Rev. Let.*, 100(4):041601, 2008.
5. J. Brannick, Y. Chen, X. Hu, and L. Zikatanov. Parallel unsmoothed aggregation algebraic multigrid algorithms on gpus. In *Numerical Solution of Partial Differential Equations: Theory, Algorithms and their Applications, Springer Series in Mathematics and Statistics*. Springer, Berlin/Heidelberg, Accepted 2012.
6. J. Brannick, Y. Chen, J. Krauss, and L. Zikatanov. An algebraic multigrid method based on matching of graphs. *Lecture notes in Computational Science and Engineering*, 2012.
7. J. Brannick, Y. Chen, and L. Zikatanov. An algebraic multilevel method for anisotropic elliptic equations based on subgraph matching. *Numer. Linear Algebra Appl.*, 19:279–295, 2012.
8. M. Brezina, P.Vaněk, and P. Vassilevski. An improved convergence analysis of smoothed aggregation algebraic multigrid. *Journal of Numerical Linear Algebra and Applications*, 19:pages 441–469, 2012.
9. J. Cheeeger. A lower bound for the smallest eigenvalue of the Laplacian. In Robert C. Gunning, editor, *Problems in Analysis, A Symposium in Honor of Salomon Bochner*, pages 195–199. Princeton Univ. Press, New Jersey, 1970.
10. J. Kraus, P. Vassilevski, and L. Zikatanov. Polynomial of best uniform approximation to x^{-1} and smoothing in two-level methods. *Comput. Methods Appl. Math.*, 12:448–468, 2012. Also available at arXiv:1002.1859v3 [math.NA].
11. O.E. Livne and A. Brandt. Lean algebraic multigrid (LAMG): Fast graph Laplacian linear solver. *SIAM Journal of Scientific Computing*, 34:B499–B522, 2012.
12. A. Napov and Y. Notay. An algebraic multigrid method with guaranteed convergence rate. *SIAM J. Sci. Comput.*, 34:A1079A1109, 2012.
13. P. Vaněk, M. Brezina, and J. Mandel. Convergence of algebraic multigrid based on smoothed aggregation. *Numer. Math.*, 88:559–579, 2001.

Space-Time Domain Decomposition for Mixed Formulations of Diffusion Equations

Thi-Thao-Phuong Hoang[1], Jérôme Jaffré[1], Caroline Japhet[1,2], Michel Kern[1] and Jean Roberts[1]

1 Introduction

Flow and transport problems in porous media are well-known for their high computational cost. In the far field simulation of an underground nuclear waste disposal site, one has to work with extremely different length and time scales, and highly variable coefficients while satisfying strict accuracy requirements. One strategy for tackling these difficulties is to apply a non-overlapping domain decomposition method which allows local adaptation in both space and time and makes possible the use of parallel algorithms. The substructuring method with a Steklov Poincaré operator, which is widely used by engineers for steady problems with strong heterogeneities, is a promising option. The optimized Schwarz waveform relaxation (OSWR) method, which has been developed over the last decade for finite element and finite volume methods, is another potential choice.
The objective of this paper is twofold. Firstly, we propose the time-dependent Steklov Poincaré operator and introduce the Neumann-Neumann preconditioner [2] as well as the weight matrices [13] to improve the convergence speed and handle the heterogeneities. Secondly, we extend the OSWR approach [8] to the case of mixed finite elements [3] with their local mass-conservation property. Numerical experiments in 2D are presented to illustrate the performance of the two methods for a simplified ANDRA test case.

For an open, bounded subset Ω of $\mathbb{R}^d$ $(d = 2,3)$ with Lipschitz boundary $\partial\Omega$ and some fixed time $T > 0$, we consider the following time-dependent diffusion problem

$$\omega\partial_t c + \nabla\cdot(-\mathbf{D}\nabla c) = f \text{ in } \Omega\times(0,T), \tag{1}$$

$$c = 0 \text{ on } \partial\Omega\times(0,T), \tag{2}$$

$$c(0,\cdot) = c_0 \text{ in } \Omega, \tag{3}$$

where c is the concentration of a contaminant, ω the porosity and $\mathbf{D}$ a symmetric, positive definite diffusion tensor.
We now rewrite (1) in an equivalent mixed form by introducing the vector field $\mathbf{r} := -\mathbf{D}\nabla c$. This yields

$$\begin{aligned} \omega\partial_t c + \nabla\cdot\mathbf{r} &= f \text{ in } \Omega\times(0,T), \\ \mathbf{D}^{-1}\mathbf{r} + \nabla c &= 0 \text{ in } \Omega\times(0,T), \end{aligned} \tag{4}$$

[1] INRIA, Rocquencourt, France, e-mail: {Phuong.Hoang_Thi_Thao}{Jerome.Jaffre}{Michel.Kern}{Jean.Roberts}@inria.fr .[2] Université Paris 13, LAGA, Villetaneuse, France, e-mail: japhet@math.univ-paris13.fr

along with boundary and initial conditions (2) - (3). Henceforth, unless otherwise specified, we implicitly assume boundary condition (2) on $\partial\Omega$.

Theorem 1. (Well-posedness and Regularity)

Suppose that the diffusion tensor $\mathbf{D}$ *is in* $W^{1,\infty}(\Omega)^{d^2}$. *If* $f \in L^2(0,T;L^2(\Omega))$ *and* $c_0 \in H_0^1(\Omega)$, *then problem (4) has a unique weak solution* $(c,\mathbf{r})$ *such that*

$$c \in H^1(0,T;L^2(\Omega)) \text{ and } \mathbf{r} \in L^2(0,T;H(\mathrm{div},\Omega)) \cap L^\infty(0,T;L^2(\Omega)^d).$$

Moreover, if $f \in H^1(0,T;L^2(\Omega))$ *and* $c_0 \in H^2(\Omega) \cap H_0^1(\Omega)$ *then*

$$c \in W^{1,\infty}(0,T;L^2(\Omega)) \text{ and } \mathbf{r} \in L^\infty(0,T;H(\mathrm{div},\Omega)) \cap H^1(0,T;L^2(\Omega)^d).$$

The proof is based on energy estimates and Galerkin's method (see [12, 9]).

2 Two space-time domain decomposition methods

Our work relies on the decomposition of Ω into non-overlapping subdomains. For simplicity, we describe the methods in case of two non-overlapping subdomains Ω_1 and Ω_2 with $\Gamma = \partial\Omega_1 \cap \partial\Omega_2 \cap \Omega$ (the results can be extended to the case of many subdomains as we shall see in the numerical experiments).

Let $\{c_i, \mathbf{r}_i\}$ be the restriction to Ω_i, $i = 1,2$, of $\{c,\mathbf{r}\}$, the solution to (4). Problem (4) can be reformulated in the equivalent multi-domain form by solving the same problem (globally in space and time) in each subdomain:

$$\begin{aligned} \omega_i \partial_t c_i + \nabla\cdot\mathbf{r}_i &= f \quad \text{in } \Omega_i \times (0,T) \\ \mathbf{D}_i^{-1}\mathbf{r}_i + \nabla c_i &= 0 \quad \text{in } \Omega_i \times (0,T) \qquad \text{for } i = 1,2, \\ c_i(0) &= c_0 \text{ in } \Omega_i \end{aligned} \tag{5}$$

along with the physical transmission conditions on the space-time interface

$$\begin{aligned} c_1 &= c_2 \\ \mathbf{r}_1\cdot\mathbf{n}_1 + \mathbf{r}_2\cdot\mathbf{n}_2 &= 0 \end{aligned} \qquad \text{on } \Gamma\times(0,T). \tag{6}$$

where $\mathbf{n}_i$ is the outward unit normal vector on $\partial\Omega_i$.

2.1 Method 1: Time-dependent Steklov-Poincaré operator approach

This method is the continuous counterpart of the Schur complement method, but extended to the time-dependent problem.

For f and c_0 as before and $\lambda \in L^2(0,T;H^{\frac{1}{2}}(\Gamma))$, we define the extension operators

$$\mathscr{D}_i : (\lambda, f, c_0) \mapsto (c_i(\lambda, f, c_0), \mathbf{r}_i(\lambda, f, c_0)),$$

where $(c_i(\lambda, f, c_0), \mathbf{r}_i(\lambda, f, c_0))$, $i = 1,2$, is the solution to the problem

$$\begin{aligned} \omega_i \partial_t c_i + \nabla\cdot\mathbf{r}_i &= f \quad \text{in } \Omega_i \times (0,T), \\ \mathbf{D}_i^{-1}\mathbf{r}_i + \nabla c_i &= 0 \quad \text{in } \Omega_i \times (0,T), \\ c_i &= \lambda \quad \text{on } \Gamma \times (0,T), \\ c_i(0) &= c_0 \text{ in } \Omega_i. \end{aligned} \tag{7}$$

Comparing with (5), (6), $(c_i(\lambda, f, c_0), \mathbf{r}_i(\lambda, f, c_0))$ satisfies (5) - (6) if and only if

$$\mathbf{r}_1(\lambda, f, c_0)\cdot\mathbf{n}_1 + \mathbf{r}_2(\lambda, f, c_0)\cdot\mathbf{n}_2 = 0 \quad \text{on } \Gamma\times(0,T),$$

or equivalently,

$$\mathscr{F}_1\mathscr{D}_1(\lambda, f, c_0) + \mathscr{F}_2\mathscr{D}_2(\lambda, f, c_0) = 0 \quad \text{on } \Gamma\times(0,T), \tag{8}$$

where $\mathscr{F}_i(c_i,\mathbf{r}_i) := \mathbf{r}_i\cdot\mathbf{n}_i\,|_\Gamma$, $i=1,2$, is the normal trace operator.
As the operators $\mathscr{F}_i$ and $\mathscr{D}_i$ are affine in λ, (8) can be rewritten as

$$\mathscr{S}\lambda = \chi \qquad \text{on } \Gamma\times(0,T), \tag{9}$$

where $\mathscr{S}$ is the linear time-dependent Steklov-Poincaré operator, defined by

$$\mathscr{S} = \mathscr{S}_1 + \mathscr{S}_2,\ \mathscr{S}_i\lambda := -\mathscr{F}_i\mathscr{D}_i(\lambda,0,0) \text{ (Dirichlet-to-Neumann operator).}$$

And the right-hand side is

$$\chi = \mathscr{F}_1\mathscr{D}_1(0, f, c_0) + \mathscr{F}_2\mathscr{D}_2(0, f, c_0).$$

Remark.

(i) Subdomain problem (7) is wellposed (this is an easy extension of Theorem 1).

(ii) We solve problem (9) iteratively using a Krylov subspace method such as GMRES.

(ii) The operator $\mathscr{S}$ is non-symmetric. In particular, by writing the variational formulations of the subdomain problems, we deduce for $\lambda,\eta\in L^2(0,T;H^{\frac{1}{2}}(\Gamma))$ that

$$\langle\mathscr{S}\lambda,\eta\rangle = \sum_{i=1}^{2}\left(\int_0^T\int_{\Omega_i}\mathbf{D}^{-1}\tilde{\mathbf{r}}_i(\eta)\cdot\tilde{\mathbf{r}}_i(\lambda) + \int_0^T\int_{\Omega_i}\omega_i\frac{\partial\tilde{c}_i(\lambda)}{\partial t}\tilde{c}_i(\eta)\right),$$

where $(\tilde{c}_i(\lambda),\tilde{\mathbf{r}}_i(\lambda)) := \mathscr{D}_i(\lambda,0,0)$ for $i=1,2$. Thus, the well-posedness of (9) is still an open question (see a related work by F. Kwok [11]).

2.2 *Method 2: Optimized Schwarz waveform relaxation approach*

We consider the second domain decomposition approach, the Optimized Schwarz Waveform Relaxation (OSWR) method, where we replace the physical transmission conditions (6) by the equivalent Robin conditions on the space-time interface

$$\begin{aligned} -\mathbf{r}_1\cdot\mathbf{n}_1 + p_1c_1 &= -\mathbf{r}_2\cdot\mathbf{n}_1 + p_1c_2 \\ -\mathbf{r}_2\cdot\mathbf{n}_2 + p_2c_2 &= -\mathbf{r}_1\cdot\mathbf{n}_2 + p_2c_1 \end{aligned} \qquad \text{on } \Gamma\times(0,T), \tag{10}$$

where p_1 and p_2 are positive parameters that can be optimized to significantly improve the convergence rate of the method (see [1, 4, 5] and the references therein).

The OSWR method may be written as follows: at the k^{th} iteration, we solve in each subdomain the problem

$$\begin{aligned} \partial_t c_i^k + \nabla\cdot\mathbf{r}_i^k &= f && \text{in } \Omega_i\times(0,T), \\ \mathbf{D}_i^{-1}\mathbf{r}_i^k + \nabla c_i^k &= 0 && \text{in } \Omega_i\times(0,T), \\ -\mathbf{r}_i^k\cdot\mathbf{n}_i + p_ic_i^k &= -\mathbf{r}_j^{k-1}\cdot\mathbf{n}_i + p_ic_j^{k-1} && \text{on } \Gamma\times(0,T),\ j=(3-i), \\ c_i(0,\cdot) &= c_0 && \text{in } \Omega_i. \end{aligned} \tag{11}$$

Remark.

(i) For the first iteration, the transmission conditions are replaced by

$$-\mathbf{r}_i^1 \cdot \mathbf{n}_i + p_i c_i^1 = g_i, \quad \text{on } \Gamma \times (0,T)$$

for $g_i, i = 1,2$, an initial guess on the space-time interface.

(ii) The well-posedness of subdomain problem (11) is an extension of Theorem 1 (see [9]) making use of the space $\mathscr{H}(div, \Omega_i)$ defined by

$$\mathscr{H}(\text{div}, \Omega_i) = \left\{ \mathbf{v} \in H(\text{div}, \Omega_i) \text{ such that } \mathbf{v} \cdot \mathbf{n}_i \in L^2(\Gamma) \right\}.$$

Theorem 2. (Convergence of the OSWR method in mixed form)

Suppose that $\mathbf{D}$ *is in* $W^{1,\infty}(\Omega)^{d^2}$. *Let* $f \in H^1(0,T;L^2(\Omega))$ *and* $c_0 \in H^2(\Omega) \cap H_0^1(\Omega)$. *If the algorithm (11) is initialized by* (g_i) *given in* $H^1\left(0,T;L^2(\Gamma)\right)$, *then it defines a unique sequence of iterates*

$$(c_i^k, \mathbf{r}_i^k) \in W^{1,\infty}(0,T;L^2(\Omega_i)) \times L^\infty(0,T;\mathscr{H}(\text{div}, \Omega_i)) \cap H^1(0,T;L^2(\Omega_i)^d),\ i = 1,2,$$

that converges to the weak solution $(c, \mathbf{r})$ *of problem (4).*

Remark. Theorem 2 can be extended to the case of many subdomains (see [9]).

As in subsection 2.1, we now derive an interface problem. However, here we use two interface unknowns: let $\zeta_i \in H^1\left(0,T;L^2(\Gamma)\right), i = 1,2$. We define the following extension operators:

$$\mathscr{R}_i : (\zeta_i, f, c_0) \mapsto (c_i(\zeta_i, f, c_0), \mathbf{r}_i(\zeta_i, f, c_0)), \tag{12}$$

where $(c_i(\zeta_i, f, c_0), \mathbf{r}_i(\zeta_i, f, c_0))$, $i = 1,2$, is the solution to the problem

$$\begin{array}{rl} \omega_i \partial_t c_i + \nabla \cdot \mathbf{r}_i = f & \text{in } \Omega_i \times (0,T), \\ \mathbf{D}_i^{-1} \mathbf{r}_i + \nabla c_i = 0 & \text{in } \Omega_i \times (0,T), \\ -\mathbf{r}_i \cdot \mathbf{n}_i + p_i c_i = \zeta_i & \text{on } \Gamma \times (0,T), \\ c_i(0) = c_0 & \text{in } \Omega_i. \end{array} \tag{13}$$

The interface operators are denoted by $\mathscr{B}_i$, $i = 1,2$, and are defined by

$$\mathscr{B}_i(c_j, \mathbf{r}_j) = (-\mathbf{r}_j \cdot \mathbf{n}_i + p_i c_j)\,|_\Gamma,\ j = (3-i). \tag{14}$$

Thus, transmission conditions (10) lead to the interface problem

$$\begin{array}{l} \zeta_1 = \mathscr{B}_1 \mathscr{R}_2(\zeta_2, f, c_0) \\ \zeta_2 = \mathscr{B}_2 \mathscr{R}_1(\zeta_1, f, c_0) \end{array} \quad \text{on } \Gamma \times (0,T), \tag{15}$$

or equivalently,

$$\begin{pmatrix} I & -\mathscr{B}_1 \mathscr{R}_2(\cdot,0,0) \\ -\mathscr{B}_2 \mathscr{R}_1(\cdot,0,0) & I \end{pmatrix} \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix} = \begin{pmatrix} \mathscr{B}_1 \mathscr{R}_2(0,f,c_0) \\ \mathscr{B}_2 \mathscr{R}_1(0,f,c_0) \end{pmatrix} \text{ on } \Gamma \times (0,T).$$

We solve this system iteratively using Jacobi iteration (this is the OSWR method (11)) or using GMRES.

3 Discontinuous Galerkin time stepping with different subdomain time grids

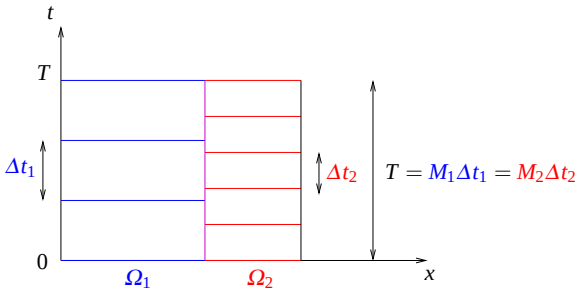


Fig. 1 Non-conforming time grids in the subdomains.

As the two methods described in the previous section are global in time, we can use different time steps in different subdomains according to their physical properties. We consider two possibly different uniform partitions $\mathscr{T}_1$ and $\mathscr{T}_2$ of the time interval $(0,T)$ into sub-intervals of lengths Δt_1 and Δt_2 respectively. We denote by J_m^i the interval $(t_{m-1}^i, t_m^i]$, $m = 1,\ldots,M_i$, for $i = 1,2$. In particular, we are interested in the non-conforming case where $\Delta t_1 \neq \Delta t_2$ as depicted in Fig. 1. For the time discretization, we use the discontinuous Galerkin method [10, 8]. In this paper, we consider the lowest order scheme, which is a modified backward Euler method. We denote by $P_0(\mathscr{T}_i, W)$ the space of piecewise constant functions in time on grid $\mathscr{T}_i$ with values in W where $W = H^{\frac{1}{2}}(\Gamma)$ for Method 1 and $W = L^2(\Gamma)$ for Method 2:

$$P_0(\mathscr{T}_i, W) = \left\{ \phi : (0,T) \to W,\ \phi \text{ is constant in time on } J_m^i,\ \forall m = 1,\ldots,M_i \right\}.$$

In order to exchange data on the space-time interface between different time grids, we define the following L^2 projection Π_{ji} from $P_0(\mathscr{T}_i, W)$ onto $P_0(\mathscr{T}_j, W)$: for $\phi \in P_0(\mathscr{T}_i, W)$, $\Pi_{ji}\phi\,|_{J_m^j}$ is the average value of ϕ on J_m^j, for $m = 1,\ldots,M_j$. We use a simple algorithm [6] for effectively performing this projection. With these tools, we are now able to weakly enforce the transmission conditions over the time intervals.

For Method 1. We take λ piecewise constant in time (on grid $\mathscr{T}_1$, or $\mathscr{T}_2$ or on yet another grid). Let, for instance, $\lambda \in P_0(\mathscr{T}_1, H^{\frac{1}{2}}(\Gamma))$. Thus, we have

$$c_1 \;=\; \Pi_{11}(\lambda) \;=\; \mathrm{Id}(\lambda) \text{ and } c_2 = \Pi_{21}(\lambda), \quad \text{on } \Gamma \times (0,T).$$

The flux is then conserved over each time interval J_m^1 by letting

$$\int_{J_m^1} \left(\Pi_{11}\left(\mathbf{r}_1(\Pi_{11}(\lambda)) \cdot \mathbf{n}_1\right) + \Pi_{12}\left(\mathbf{r}_2(\Pi_{21}(\lambda)) \cdot \mathbf{n}_2\right) \right) dt = 0, \text{ for } m = 1,\ldots,M_1.$$

For Method 2. As we have two Lagrange multipliers on the space-time interface, we take $\zeta_i \in P_0(\mathscr{T}_i, L^2(\Gamma))$ for $i = 1,2$ and enforce the conservation of the jumps of the two Robin terms over the time intervals [8] by letting

$$\int_{J_m^i} \left(\zeta_i - \Pi_{ij}\left(-\mathbf{r}_j(\zeta_j) \cdot \mathbf{n}_i + p_i c_j(\zeta_j)\right) \right) dt = 0,$$

for $m = 1, \cdots, M_i$, and for $i = 1,2$, $j = (3-i)$.

4 Numerical Experiments

We consider $2D$ problems with $\mathbf{D} = d\mathbf{I}$ isotropic and constant in each subdomain, where $\mathbf{I}$ is the identity matrix. We then denote by $d_i := d|_{\Omega_i}$.

For Method 1: Using a Neumann-Neumann Preconditioner. In the elliptic case with strong heterogeneity, the convergence of an iterative method for the Schur complement problem enhanced with a Neumann-Neumann preconditioner and weight matrices is independent of the jump in the coefficients [13]. Thus, we extend the idea to our method for parabolic problem. In particular, we rewrite the interface problem (9) as

$$\left(\delta_1 \mathscr{S}_1^{-1} + \delta_2 \mathscr{S}_2^{-1}\right)\left(\mathscr{S}_1 + \mathscr{S}_2\right)\lambda = \hat{\chi} \qquad \text{on } \Gamma \times (0,T),$$

where $\delta_i = [d_i/(d_1+d_2)]^2$ and $\mathscr{S}_i^{-1}$, the Neumann-to-Dirichlet operator, is the inverse of $\mathscr{S}_i$ for $i = 1,2$. This formula can be generalized to the case of many subdomains.

For Method 2: Using two optimization techniques. To calculate the optimized Robin parameters for discontinuous coefficients, the first approach is to optimize the convergence factor based on the two-half space Fourier analysis [4], we call this approach Opt. 1. In our application to nuclear waste problems where the geometry consists of small objects embedded in a large space, we use an adapted optimization proposed in [7], called Opt. 2, which takes into account the size of the subdomains.

We consider a test case designed by ANDRA for the pure diffusion equation. The

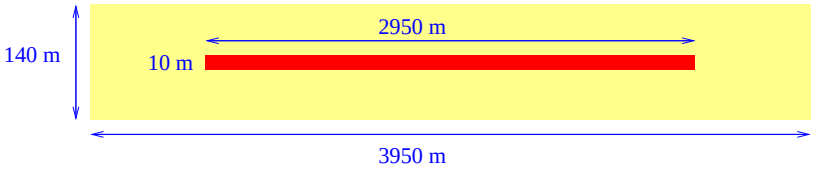


Fig. 2 Geometry of the domain.

geometry of the physical domain is depicted in Fig. 2. The porosity is $\omega = 0.2$ in the repository (in red) and $\omega = 0.05$ in the clay layer (in yellow). The diffusion coefficient is $d = 2 \times 10^{-9}$ m^2 s^{-1} in the repository and $d = 5 \times 10^{-12}$ m^2 s^{-1} in the clay layer. The source term is

$$f = \begin{cases} 10^{-5} \text{ mol/s} & \text{if } t \leq 10^5 \text{ years}, \\ 0 & \text{if } t > 10^5 \text{ years}, \end{cases}$$

in the repository, and $f = 0$ in the clay layer.

For the spatial discretization, we use a non-uniform rectangular mesh with a finer discretization in the repository (a uniform mesh with 600 points in the x direction

and 30 points in the y direction) and a coarser discretization in the clay layer (the mesh size progressively increases with distance to the repository by a factor of 1.05). We then apply mixed finite elements with the lowest order Raviart-Thomas space on rectangles. For the time discretization, we use non-matching time grids with $\Delta t = 2000$ years in the repository and $\Delta t = 10000$ years in the clay layer. Finally, we decompose the domain into 9 rectangular subdomains (3×3 with the repository represented by one subdomain).

To analyze and compare the convergence results of different algorithms, we solve a

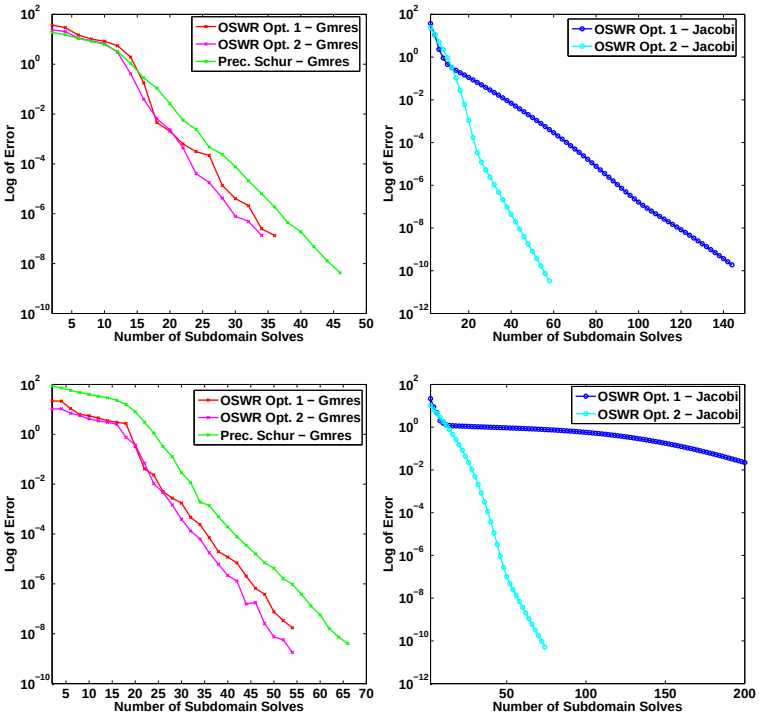


Fig. 3 Convergence curves for different algorithms and time intervals: with GMRES (on the left) and with Jacobi (on the right), for short time $T = 200,000$ years (on top) and for long time $T = 1,000,000$ years (on below).

problem with the right hand side $f \equiv 0$. We start with random initial guesses on the space-time interfaces and check the convergence to zero in $L^2(0,T;L^2(\Omega))-$norms of the concentration and vector field, with tolerance 10^{-6} on the residual. We remark that one iteration of Method 1 with the preconditioner costs twice as much as one iteration of Method 2 (in terms of number of subdomain solves). Thus we plot the error versus the number of subdomain solves (instead of versus the number of iterations). In Fig. 3, we compare the errors for different algorithms (GMRES on the left and Jacobi iteration on the right) and over different time intervals (shorter interval on top and longer interval on bottom). The same time steps, Δt_i, are used for

the shorter and longer time intervals. We observe that with GMRES, both Method 1 (with Neumann-Neumann preconditioner) and Method 2 (with either Opt. 1 or Opt. 2) work well and their performance is comparable. The convergence becomes slower when the time interval increases, which is reasonable and expected. On the other hand, with Jacobi iteration, we see that the performance of Opt. 1 (classical) is far behind Opt. 2 (adapted), especially for the long time case.

Acknowledgements This work was supported by ANDRA, the French Agency for Nuclear Waste Management.

References

1. Bennequin, D., Gander, M.J., Halpern, L.: A homographic best approximation problem with application to optimized Schwarz waveform relaxation. Math. Comput. **78**, 185–223 (2009)
2. DeRoeck, Y.H., Tallec, P.L.: Analysis and test of a local domain decomposition preconditioner. In: R. Glowinski, Y. Kuznetsov, G. Meurant, J. Périaux, O. Widlund (eds.) Proceedings of the 4th International Symposium on Domain Decomposition Methods for Partial Differential Equations. SIAM, Philadelphia, PA (1991)
3. Douglas, J., Leme, P.P., Roberts, J., Wang, J.: A parallel iterative procedure applicable to the approximate solution of second order partial differential equations by mixed finite element methods. Numer. Math. **65**, 95–108 (1993)
4. Gander, M.J., Halpern, L., Kern, M.: A Schwarz waveform relaxation method for advection-diffusion-reaction problems with discontinuous coefficients and non-matching grids. In: O. Widlund, D.E. Keyes (eds.) Decomposition Methods in Science and Engineering XVI, Heidelberg, pp. 916–920. Springer (2007)
5. Gander, M.J., Halpern, L., Nataf, F.: Optimized Schwarz waveform relaxation method for the one dimensional wave equation. SIAM J. Numer. Anal. **41(5)** (2003)
6. Gander, M.J., Japhet, C., Maday, Y., Nataf, F.: A new cement to glue non-conforming grids with Robin interface conditions: The finite element case. In: R. Kornhuber, R.H.W. Hoppe, J. Périaux, O. Pironneau, O.B. Widlund, J. Xu (eds.) Proceedings of the 5th International Conference on Domain Decomposition Methods, vol. 40, pp. 259–266. Springer LNCSE (2005)
7. Halpern, L., Japhet, C., Omnes, P.: Nonconforming in time domain decomposition method for porous media applications. In: J.C.F. Pereira, A. Sequeira (eds.) Proceedings of the 5th European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010. Lisbon, Portugal (2010)
8. Halpern, L., Japhet, C., Szeftel, J.: Discontinuous Galerkin and nonconforming in time optimized Schwarz waveform relaxation. In: Y. Huang, R. Kornhuber, O. Widlund, J. Xu (eds.) Domain Decomposition Methods in Science and Engineering XIX, pp. 133–140 (2011)
9. Hoang, T.T.P., Jaffré, J., Japhet, C., Kern, M., Roberts, J.E.: Domain decomposition methods for time-dependent diffusion problems in mixed formulations. Research report, N^o 8271, HAL: inria-00803796 (http://hal.inria.fr/hal-00803796/) (2013)
10. Johnson, C., Eriksson, K., Thomée, V.: Time discretization of parabolic problems by discontinuous Galerkin method. RAIRO Modél. Math. Anal. Numér. **19** (1995)
11. Kwok, F.: Neumann-Neumann waveform relaxation for the time-dependent heat equation. In: This proceedings
12. Li, J., Arbogast, T., Huang, Y.: Mixed methods using standard conforming finite elements. Comput. Methods Appl. Mech. Engrg. **198**, 680–692 (2009)
13. Mandel, J., Brezina, M.: Balancing domain decomposition for problems with large jumps in coefficients. Math. Comput. **65**, 1387–1401 (1996)

Block Jacobi for discontinuous Galerkin discretizations: no ordinary Schwarz methods

Martin J. Gander[1] and Soheil Hajian[1]

1 Introduction

We study in this paper block Jacobi iterations for matrix problems obtained by discontinuous Galerkin (DG) discretizations. To fix ideas, we consider the model problem

$$\begin{aligned} -\Delta u &= f, \quad \text{in } \Omega \subset \mathbb{R}^2, \\ u &= 0, \quad \text{on } \partial\Omega. \end{aligned} \tag{1}$$

Any discretization of (1) leads to a linear system of equations of the form

$$A\mathbf{y} = \mathbf{f}, \tag{2}$$

where $\mathbf{y}$ is the vector of degrees of freedom representing approximations of u and possibly ∇u. A block Jacobi iteration with two non-overlapping subblocks is given by

$$M\mathbf{y}^{(n+1)} = N\mathbf{y}^{(n)} + \mathbf{f}, \quad M = \left[\begin{array}{c|c} A_1 & O \\ \hline O & A_2 \end{array}\right], \; N = -\left[\begin{array}{c|c} O & A_{12} \\ \hline A_{21} & O \end{array}\right]. \tag{3}$$

For classical discretizations of elliptic partial differential equations, like conforming finite elements or finite differences, block Jacobi methods are equivalent to classical Schwarz methods with minimal overlap, see for example [4]. This is different when the linear system (1) is obtained using DG methods.

Our paper is organized as follows: in section 2 we describe several DG methods for linear elliptic problems. We follow our discussion by introducing some "hybridizable" DG methods. In section 3 we show that block Jacobi iterations for the DG methods are corresponding to non-overlapping Schwarz methods with particular transmission conditions involving the penalty parameter of the DG method used. We then show numerical experiments in section 4, and present our conclusions in section 5.

2 Discontinuous Galerkin methods

We introduce the so-called flux formulations, which define a class of discontinuous Galerkin methods for linear elliptic problems. We use the unified framework presented in [1].

[1] Université de Genève, 2-4 rue du Lièvre, CP 64, CH-1211 Genève 4,e-mail: {Martin.Gander,Soheil.Hajian}@unige.ch

Let $\mathscr{T}_h = \{K\}$ be a shape-regular triangulation of a polyhedral domain $\Omega \subset \mathbb{R}^2$. Let $h = \max_{K\in\mathscr{T}_h} h_K$. We denote by $\mathscr{E}^0$ the set of interior edges shared by all $K \in \mathscr{T}_h$, the set of boundary edges $\mathscr{E}^\partial$ and all edges by $\mathscr{E} := \mathscr{E}^\partial \cup \mathscr{E}^0$.

Following [1] we define the broken Sobolev space $H^l(\mathscr{T}_h) := \prod_{K\in\mathscr{T}_h} H^l(K)$ and the trace space $T(\mathscr{E}) = \prod_{K\in\mathscr{T}_h} L^2(\partial K)$ where $H^l(K)$ is the Sobolev space in $K \in \mathscr{T}_h$. We also define two trace operators: let $q \in T(\mathscr{E})$ and $\varphi \in [T(\mathscr{E})]^2$. On $e = \partial K_1 \cap \partial K_2$ we then define average $\{\{\cdot\}\}$ and jump $[\![\cdot]\!]$ operators by

$$\begin{array}{ll} \{\{q\}\} = \frac{1}{2}(q_1+q_2), & [\![q]\!] = q_1 n_1 + q_2 n_2, \\ \{\{\varphi\}\} = \frac{1}{2}(\varphi_1+\varphi_2), & [\![\varphi]\!] = \varphi_1 \cdot n_1 + \varphi_2 \cdot n_2, \end{array} \tag{4}$$

where n_i is the outward normal of K_i on e, $q_i := q|_{\partial K_i \cap e}$ and $\varphi_i := \varphi|_{\partial K_i \cap e}$. On the boundary of Ω we set the average and jump operators to be $\{\{\varphi\}\} = \varphi$ and $[\![q]\!] = q\,n$ respectively. We do not need to define $\{\{q\}\}$ and $[\![\varphi]\!]$ on $e \in \mathscr{E}^\partial$; see [1].

We denote two finite dimensional broken spaces on $\mathscr{T}_h$ for the discrete approximation by $V_h := \left\{ v \in L^2(\Omega) \text{ s.t. } v|_K \in P(K), \forall K \in \mathscr{T}_h \right\}$ where $P(K) = \mathbb{P}_k(K)$ and $\Sigma_h := \left\{ \tau \in [L^2(\Omega)]^2 \text{ s.t. } \tau|_K \in \Sigma(K), \forall K \in \mathscr{T}_h \right\}$ where $\Sigma(K) = [\mathbb{P}_k(K)]^2$. Here $\mathbb{P}_k(K)$ is the space of polynomials of degree $\leq k$ in the simplex $K \in \mathscr{T}_h$.

For the sake of simplicity we denote the volume and surface integrals by $(a,b)_K = \int_K a\,b$ for $K \in \mathscr{T}_h$ and $\langle a,b\rangle_e = \int_e a\,b$ for $e \in \mathscr{E}$. Moreover $\|v\|^2_{0,\mathscr{T}_h} := \sum_{K\in\mathscr{T}_h} (v,v)_K$.

2.1 Flux formulation

For the Laplacian model problem (1) in the DG context, one first rewrites the equation in mixed form,

$$\sigma = \nabla u, \quad -\nabla\cdot\sigma = f(x), \quad x \in \Omega. \tag{5}$$

Then the *flux formulation* is the following: let $K \in \mathscr{T}_h$, $v \in P(K)$ and $\tau \in \Sigma(K)$. We multiply (5) by τ and v respectively. Integrating by parts over K, we substitute boundary terms of u and σ by two approximation functions. Hence the discrete weak form reads: find $(u_h, \sigma_h) \in V_h \times \Sigma_h$ for all $K \in \mathscr{T}_h$ such that

$$\begin{array}{ll} (\sigma_h, \tau)_K = -(u_h, \nabla\cdot\tau)_K + \langle \hat{u}_h, \tau\cdot n_K\rangle_{\partial K} & \forall \tau \in \Sigma(K), \\ (\sigma_h, \nabla v)_K = (f,v)_K \langle v, \hat{\sigma}_h \cdot n_K\rangle_{\partial K} & \forall v \in P(K), \end{array} \tag{6}$$

where n_K is the outward normal of element K and

$$\hat{u}_h : H^2(\mathscr{T}_h) \times \left[H^1(\mathscr{T}_h)\right]^2 \to T(\mathscr{E}), \quad \hat{\sigma}_h : H^2(\mathscr{T}_h) \times \left[H^1(\mathscr{T}_h)\right]^2 \to [T(\mathscr{E})]^2, \tag{7}$$

which are called numerical fluxes. They approximate the traces of u_h and σ_h on ∂K. By defining $\hat{u}_h$ and $\hat{\sigma}_h$ we complete the definition of a DG method.

For instance we introduce the *local discontinuous Galerkin* method (LDG) with

$$\begin{aligned} \hat{u}_h &= \{\{u_h\}\} - \beta \cdot [\![u_h]\!] \text{ on } \mathscr{E}^0, \quad \hat{u}_h = 0 \text{ on } \partial\Omega, \\ \hat{\sigma}_h &= \{\{\sigma_h\}\} + \beta\, [\![\sigma_h]\!] - \mu [\![u_h]\!] \text{ on } \mathscr{E}, \end{aligned} \tag{8}$$

where $\beta \in \left[L^2(\mathscr{E})\right]^2$ is a constant vector-valued function with $\beta = 0$ on $\partial\Omega$ and $\mu \propto h_e^{-1}$ where h_e is the edge length. We will consider the case $\beta = -n_{K_1}/2$ on $e = \partial K_1 \cap \partial K_2$ where $K_1, K_2 \in \mathscr{T}_h$ and the assignment of n_{K_1} is arbitrary. Therefore the numerical fluxes are

$$\hat{u}_h = (u_h)_{K_1}, \quad \hat{\sigma}_h = (\sigma_h)_{K_2} - \mu [\![u_h]\!] \text{ on } e. \tag{9}$$

In case we have non-homogeneous Dirichlet data, e.g. $u = g_D$ on $\partial\Omega$, the numerical fluxes are

$$\hat{u}_h = g_D, \quad \hat{\sigma}_h = \sigma_h - \mu\,(u_h - g_D) \text{ on } e \in \mathscr{E}^\partial. \tag{10}$$

We now introduce two more methods which are "hybridizable". A *hybrid* method is defined by eliminating interior unknowns within an element $K \in \mathscr{T}_h$ in terms of some unknowns defined on $\mathscr{E}^0$, called λ_h (which here is $\hat{u}_h$). We then obtain a system for λ_h which is much smaller than the original system. We do not derive these type of DG methods here but for a unified approach we refer the reader to [2].

Remark 1. A "hybridizable" DG method is designed to approximate the following continuous problem using $\hat{u}_h$ as Dirichlet data on ∂K:

$$\sigma - \nabla u = 0 \text{ and } -\nabla\cdot\sigma = f \text{ in } K, \quad u = \hat{u}_h(u,\sigma) \text{ on } \partial K. \tag{11}$$

More precisely, their numerical fluxes are such that $\hat{\sigma}_h = (\sigma_h)_K - \mu\,[(u_h)_K - \hat{u}_h]$ on ∂K which is the numerical flux one uses to impose Dirichlet boundary data on the boundary of an element; compare with (10).

We introduce two hybridizable methods, namely LDG-H and IP-H, by defining their numerical fluxes. The LDG-H uses

$$\begin{aligned} \hat{u}_h &= \tfrac{\mu_1}{\mu_1+\mu_2} u_{h,1} + \tfrac{\mu_2}{\mu_1+\mu_2} u_{h,2} - \tfrac{1}{\mu_1+\mu_2} [\![\sigma_h]\!], \\ \hat{\sigma}_h &= \tfrac{\mu_2}{\mu_1+\mu_2} \sigma_{h,1} + \tfrac{\mu_1}{\mu_1+\mu_2} \sigma_{h,2} - \tfrac{\mu_1\mu_2}{\mu_1+\mu_2} [\![u_h]\!], \end{aligned} \tag{12}$$

where $\mu \in T(\mathscr{E})$. Similarly for IP-H we have

$$\begin{aligned} \hat{u}_h &= \tfrac{\mu_1}{\mu_1+\mu_2} u_{h,1} + \tfrac{\mu_2}{\mu_1+\mu_2} u_{h,2} - \tfrac{1}{\mu_1+\mu_2} [\![\nabla u_h]\!], \\ \hat{\sigma}_h &= \tfrac{\mu_2}{\mu_1+\mu_2} \nabla u_{h,1} + \tfrac{\mu_1}{\mu_1+\mu_2} \nabla u_{h,2} - \tfrac{\mu_1\mu_2}{\mu_1+\mu_2} [\![u_h]\!]. \end{aligned} \tag{13}$$

One can show that IP-H and LDG-H satisfy Remark 1 by noting that for $K \in \mathscr{T}_h$

$$\hat{\sigma}_h = (\sigma_h)_K - \mu\,[(u_h)_K - \hat{u}_h] \text{ on } \partial K. \tag{14}$$

3 Domain decomposition for "hybridizable" DG methods

We decompose the domain Ω into two non-overlapping subdomains, $\{\Omega_1,\Omega_2\}$, such that the interface $\Gamma^I := \overline{\Omega}_1 \cap \overline{\Omega}_2$ is a subset of $\mathscr{E}^0$, i.e. the cut does not go through any element of $\mathscr{T}_h$. Therefore we obtain $\mathscr{T}_{h,1},\mathscr{T}_{h,2}$ from the original $\mathscr{T}_h$, and similarly $\mathscr{E}_1^0,\mathscr{E}_2^0$, for our subdomains; see for example Fig. 1 (right).

Let (u_h,σ_h) be the approximate solution obtained from a DG method. Let $(u_{h,1},\sigma_{h,1})$ be the restriction of (u_h,σ_h) to Ω_1 and similarly $(u_{h,2},\sigma_{h,2})$ to Ω_2. Then $(u_{h,i},\sigma_{h,i})$ for $i=1,2$ and $K\in\mathscr{T}_{h,i}$ satisfy

$$\begin{cases} \big(\sigma_{h,i},\tau\big)_K = -\big(u_{h,i},\nabla\cdot\tau\big)_K + \big\langle \hat{u}_{h,i},\tau\cdot n_K\big\rangle_{\partial K} & \forall \tau\in\Sigma(K),\\[2mm] \big(\sigma_{h,i},\nabla v\big)_K = (f,v)_K + \big\langle v,\hat{\sigma}_{h,i}\cdot n_K\big\rangle_{\partial K} & \forall v\in P(K), \end{cases} \tag{15}$$

where

$$\hat{u}_{h,i} := \begin{cases} \hat{u}_h(u_{h,i},\sigma_{h,i},u_{h,j},\sigma_{h,j}) & \text{on } \Gamma^I \text{ and } j\neq i,\\ \hat{u}_h(u_{h,i},\sigma_{h,i}) & \text{on } \mathscr{E}_i^0, \end{cases} \tag{16}$$

and similarly for $\hat{\sigma}_{h,i}$. Note that we do not need to define $\hat{u}_{h,1}$ on $\mathscr{E}_2^0$ since for $(u_{h,1},\sigma_{h,1})$ we only have one term in (15) that needs the trace of $(u_{h,2},\sigma_{h,2})$ on Γ^I and not $\mathscr{E}_2^0$ (similarly $\hat{u}_{h,2}$ does not need to be defined on $\mathscr{E}_1^0$).

If the trace of $(u_{h,2},\sigma_{h,2})$ is known on Γ^I, one can solve for $(u_{h,1},\sigma_{h,1})$ in Ω_1, and vice versa. This suggests an iterative algorithm for solving $(u_{h,i},\sigma_{h,i})$ in parallel, namely: find $(u_{h,i}^{(n+1)},\sigma_{h,i}^{(n+1)})$ for $i=1,2$ such that it satisfies (15) with

$$\hat{u}_{h,i} := \begin{cases} \hat{u}_h(u_{h,i}^{(n+1)},\sigma_{h,i}^{(n+1)},u_{h,j}^{(n)},\sigma_{h,j}^{(n)}) & \text{on } \Gamma^I \text{ and } j\neq i,\\ \hat{u}_h(u_{h,i}^{(n+1)},\sigma_{h,i}^{(n+1)}) & \text{on } \mathscr{E}_i^0, \end{cases} \tag{17}$$

starting with an initial guess $(u_{h,i}^{(0)},\sigma_{h,i}^{(0)})$, $i=1,2$. Note that $\hat{u}_{h,1}$ is is not equal any more to $\hat{u}_{h,2}$ on Γ^I except at convergence, and then we have $(u_{h,i}^\star,\sigma_{h,i}^\star)=(u_{h,i},\sigma_{h,i})$, i.e. the domain decomposition approximation at convergence is equal to the mono domain approximate solution.

Denoting the degrees of freedom associated with $(u_{h,i}^{(n+1)},\sigma_{h,i}^{(n+1)})$ by $y_i^{(n+1)} = (\underline{u_i}^{(n+1)},\underline{\sigma_i}^{(n+1)})^T$ after choosing a basis for $P(K)$ and $\Sigma(K)$, we can write the equivalent linear systems for our iterative method as

$$A_1 y_1^{(n+1)} = -A_{12}y_2^{(n)} + f_1, \quad A_2 y_2^{(n+1)} = -A_{21}y_1^{(n)} + f_2, \tag{18}$$

where A_{12} is obtained from $\big\langle \hat{u}_{h,1},\tau\cdot n_K\big\rangle_e, \big\langle \hat{\sigma}_{h,1}\cdot n_K, v\big\rangle_e$ for $e\subset\Gamma^I$ and A_1 is the stiffness matrix obtained from (15) in Ω_1, and similarly for Ω_2. Setting $y^{(n+1)} := (y_1^{(n+1)},y_2^{(n+1)})^T$ and $f := (f_1,f_2)^T$, we obtain precisely a block Jacobi iteration of the form (3).

For the classical finite element method with $\mathbb{P}_1$ approximation a block Jacobi iteration corresponds to a Schwarz method with minimal overlap and Dirichlet trans-

mission conditions [4]. We show now that for hybridizable DG methods the block Jacobi iteration corresponds to a general Schwarz method of the form

$$\begin{array}{llll} -\Delta u_1^{(n+1)} = f & \text{in } \Omega_1, & -\Delta u_2^{(n+1)} = f & \text{in } \Omega_2, \\ \mathscr{B}_1 u_1^{(n+1)} = \mathscr{B}_1 u_2^{(n)} & \text{on } \Gamma^I, & \mathscr{B}_2 u_2^{(n+1)} = \mathscr{B}_2 u_1^{(n)} & \text{on } \Gamma^I, \end{array} \tag{19}$$

where $\mathscr{B}_1$ and $\mathscr{B}_2$ are two linear operators determined by the particular choice of DG discretization. The following propositions show the transmission condition on Γ^I in (19), when M and N in (3) are obtained from LDG-H, IP-H and minimal dissipation LDG methods.

Proposition 1. *Let $K_1 \in \mathscr{T}_{h,1}$, $K_2 \in \mathscr{T}_{h,2}$ and $e = \overline{K}_1 \cap \overline{K}_2 \subset \Gamma^I$. If M and N in (3) are obtained from an LDG-H discretization, then the block Jacobi iteration (3) is the discrete version of (19) with $\mathscr{B}_1 = \partial_{n_1} + \mu_2$ and $\mathscr{B}_2 = \partial_{n_2} + \mu_1$ on e.*

Proof. We start with K_1: since the numerical fluxes of the LDG-H satisfy the condition in Remark 1, i.e. $\hat{\sigma}_{h,1} = \sigma_{h,1}^{(n+1)} - \mu_1(u_{h,1}^{(n+1)} - \hat{u}_{h,1})n_1$, one can conclude that we are imposing the following Dirichlet data at the continuous level: $u_1^{(n+1)} = \hat{u}_{h,1}(u_1^{(n+1)}, \sigma_1^{(n+1)}, u_2^{(n)}, \sigma_2^{(n)})$ on e. From the definition of the LDG-H numerical flux (12) we obtain

$$u_1^{(n+1)} = \frac{\mu_1}{\mu_1+\mu_2} u_1^{(n+1)} + \frac{\mu_2}{\mu_1+\mu_2} u_2^{(n)} - \frac{1}{\mu_1+\mu_2}(\sigma_1^{(n+1)} - \sigma_2^{(n)}) \cdot n_1. \tag{20}$$

Collecting terms with super index $(n+1)$ and noting $\sigma_i \cdot n_1 = \partial_{n_1} u_i$ on e, we obtain $\mathscr{B}_1 = \partial_{n_1} + \mu_2$. The same argument applies to K_2. □

Proposition 2. *Let $K_1 \in \mathscr{T}_{h,1}$, $K_2 \in \mathscr{T}_{h,2}$ and $e = \overline{K}_1 \cap \overline{K}_2 \subset \Gamma^I$. If M and N in (3) is obtained from an IP-H discretization, then the block Jacobi iteration (3) is the discrete version of (19) with $\mathscr{B}_1 = \partial_{n_1} + \mu_2$ and $\mathscr{B}_2 = \partial_{n_2} + \mu_1$ on e.*

Proof. This result can be proved similarly to the proof of Proposition 1. □

Proposition 3. *Let $K_1 \in \mathscr{T}_{h,1}$, $K_2 \in \mathscr{T}_{h,2}$ and $e = \overline{K}_1 \cap \overline{K}_2 \subset \Gamma^I$. Let M and N in (3) be obtained from a minimal dissipation LDG and assume $\beta := -n_1/2$, then the block Jacobi iteration (3) is the discrete version of (19) with $\mathscr{B}_1 = \partial_{n_1} + \mu_2$ and $\mathscr{B}_2 = 1$ on e.*

Proof. We start with K_2: note that with this definition of β we have $\hat{u}_{h,2} = u_{h,1}^{(n)}$ and $\hat{\sigma}_{h,2} = \sigma_{h,2}^{(n+1)} - \mu_2(u_{h,2}^{(n+1)} - u_{h,1}^{(n)})n_2$. Comparing with (10), one concludes that we are imposing $u_1^{(n+1)} = u_2^{(n)}$ on e. Now for K_1 using the definition of $\hat{u}_{h,1} = u_{h,1}^{(n+1)}$ on e in the first equation of (15) one obtains:

$$\left(\sigma_{h,1}^{(n+1)} - \nabla u_{h,1}^{(n+1)}, \tau\right)_{K_1} = \left\langle \hat{u}_{h,1} - u_{h,1}^{(n+1)}, \tau \cdot n_1 \right\rangle_{\partial K_1 \setminus e} \quad \forall \tau \in \Sigma(K_1). \tag{21}$$

Choosing $\tau = \nabla v$ (since $\nabla V(K_1) \subset \Sigma(K_1)$), substituting into the second equation of (15) yields

$$\left(\nabla u_{h,1}^{(n+1)}, \nabla v\right)_{K_1} = \left\langle \hat{\sigma}_{h,1} \cdot n_1, v \right\rangle_e + (f,v)_{K_1} + \left[\left\langle \hat{u}_{h,1} - u_{h,1}^{(n+1)}, \tau \cdot n_1 \right\rangle_{\partial K_1 \setminus e} + \left\langle \hat{\sigma}_{h,1} \cdot n_1, v \right\rangle_{\partial K_1 \setminus e} \right].$$

Therefore one can conclude that the following Neumann boundary data is imposed on the interface: $\sigma_1^{(n+1)} \cdot n_1 = \hat{\sigma}_{h,1}(u_1^{(n+1)}, \sigma_1^{(n+1)}, u_2^{(n)}, \sigma_2^{(n)}) \cdot n_1$ on e. Using the definition of $\hat{\sigma}_{h,1}(.) = \sigma_2^{(n)} - \mu_2(u_1^{(n+1)} - u_2^{(n)})n_1$ and collecting terms with super index $(n+1)$ leads to $\mathscr{B}_1 = \partial_{n_1} + \mu_2$. $\square$

The results here are also applicable when a positive reaction terms is present, e.g. for $(\eta - \Delta)u = f$, $\eta > 0$, since the zeroth order term only adds a term like $\eta\,(u,v)_K$ in the mixed formulation, and thus does not change numerical fluxes.

3.1 Comments on optimized Schwarz methods for DG discretizations

One can estimate the convergence of the block Jacobi method by analyzing the convergence behavior of the equivalent algorithm at the continuous level given in (19). This has been done for a simple geometry in [5], where for the case $\mu_1 = \mu_2 =: \mu$ on Γ^I, it is shown that the "uniformly optimal" value for μ is $\mu^* = \left((K_{\min}^2 + \eta)(K_{\max}^2 + \eta)\right)^{\frac{1}{4}}$. Here $K_{\min}$ and $K_{\max}$ are the minimum and maximum frequencies that can be represented on the interface, heuristically chosen to be $K_{\min} = \pi$ and $K_{\max} = \frac{\pi}{h}$ for an interface of length one. Therefore $\mu^* \propto h^{-\frac{1}{2}}$. The contraction factor of the Fourier modes in (19) is then bounded by $\rho^* = 1 - O(\sqrt{h})$. For analysis of a discretized optimized Schwarz method using FEM see [6].

We have seen that for the DG methods presented the penalty parameter enters as Robin parameter in the equivalent continuous Schwarz method. The penalty parameter in DG methods is chosen such that it ensures coercivity of the bilinear form as well as optimal convergence of the discrete approximation to the continuous solution.

Here we would like to comment only for *LDG-H* on how to choose μ such that one obtains optimal convergence to the continuous solution and achieves fast convergence of the block Jacobi iteration at the same time. For LDG-H, μ can be chosen as $O(1)$ or $O(h^{-1})$. However using [3, Theorem 2.2], it can be shown that using $\mu \propto h^{-\frac{1}{2}}$ for a class of DG methods in which LDG-H is also included yields an optimal convergence to the continuous solution and we have the following corollary.

Corollary 1. *Let the discretization be LDG-H and consider the domain decomposition setting in section 3. Set $\mu = h^{-\frac{1}{2}}$ on Γ^I and $\mu = h^{-\alpha}$ for $0 \le \alpha \le 1$ on $\mathscr{E} \setminus \Gamma^I$. Then $\|u_h - u\|_{0,\mathscr{T}_h} \le C h^{k+1}$, i.e. optimal approximation. Moreover the contraction factor of the iterative domain decomposition method (block Jacobi), is bounded by $\rho = 1 - O(\sqrt{h})$ which cannot be improved for any other choice of μ on Γ^I.*

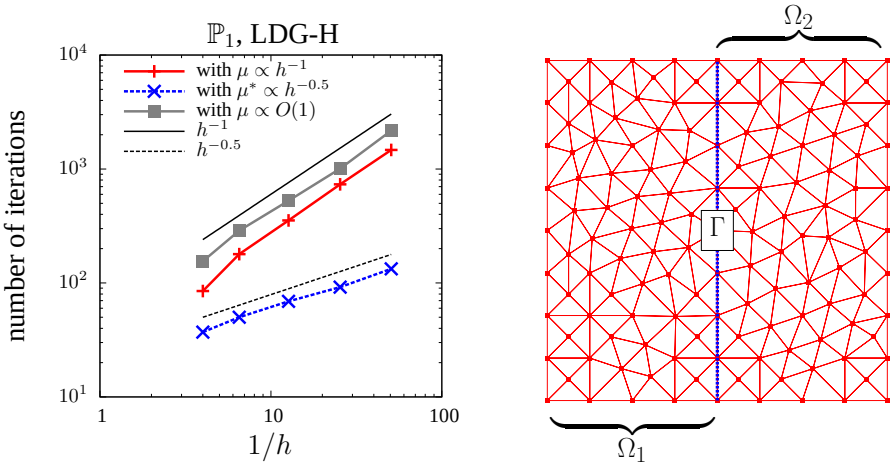


Fig. 1 (left) asymptotic number of iterations required by the non-overlapping Schwarz method using LDG-H. (right) unstructured mesh with the interface $\Gamma^I = \{0.5\} \times (0,1)$.

4 Numerical experiments

We consider $(\eta - \Delta)u = f$ in Ω and $u = 0$ on $\partial\Omega$ where we set $\eta = 1$, $\Omega = (0,1)^2$ and f such that the exact solution is $u(x,y) = \sin(\pi x)\sin(2\pi x + \frac{\pi}{4})\sin(2\pi y)$ in Ω. We illustrate the results in section 3 using a block Jacobi method as in (3) with $\Gamma^I = \{0.5\} \times (0,1)$ as interface on an unstructured mesh; see Fig. 1 (right).

The penalty parameter is usually chosen as $\mu = k^2/h_e$ where k is the degree of the polynomials; this would correspond to a very unusual high frequency approximation of the DtN operator in the optimized Schwarz method, and thus strongly affects the convergence rate. The convergence results in Fig. 2 are obtained by mea-

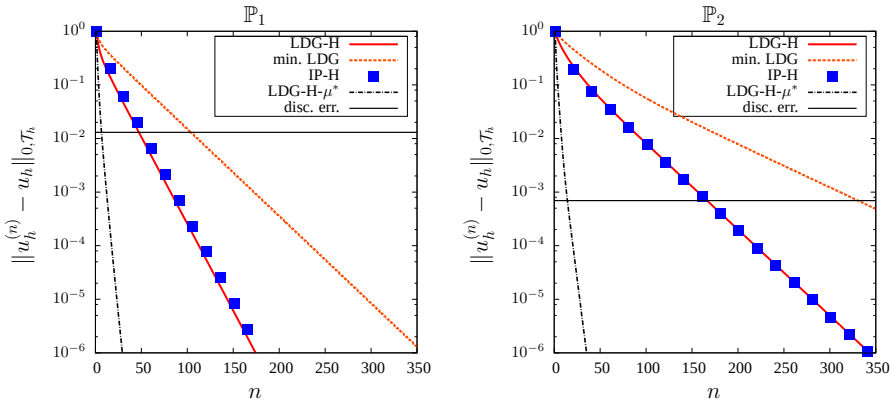


Fig. 2 Block Jacobi method for LDG-H, minimal dissipation LDG, IP-H, LDG-H with μ^* and discretization error for $\mathbb{P}_1$ and $\mathbb{P}_2$.

suring $\|u_h^{(n)} - u_h\|_{0,\mathscr{T}_h}$, where u_h is the mono-domain approximate solution and $u_h^{(n)}$ is the solution obtained at iteration n of the block Jacobi method for $\mathbb{P}_1$ and $\mathbb{P}_2$. It is evident that IP-H and LDG-H converge faster than minimal dissipation LDG in the block Jacobi iteration due to their transmission conditions. Moreover LDG-H with μ^* converges faster than LDG-H using $\mu \propto h^{-1}$ since its parameter is chosen as suggested by optimized Schwarz theory.

Fig. 1 (left) shows the number of iterations required for the block Jacobi method to reduce the iteration error to the machine precision for LDG-H with different penalty parameters on Γ^I on a sequence of unstructured meshes. We show that for LDG-H the contraction factor with "uniformly" optimal μ^* behaves as predicted in Corollary 1 and [5], i.e. $\rho^* = 1 - O(\sqrt{h})$, while with $\mu = O(1)$ or $O(h^{-1})$ behaves like $\rho = 1 - O(h)$.

5 Conclusions

We have shown that block Jacobi methods for DG discretizations correspond to non-overlapping Schwarz methods with Robin-, or Robin and Dirichlet transmission conditions. This is in contrast to standard finite element methods, where block Jacobi methods correspond to classical Schwarz methods with minimal overlap and Dirichlet transmission conditions. In addition, we found that the penalty parameter in certain DG method leads to a high frequency approximation in the transmission condition of the optimized Schwarz method, which is not a very good choice for the convergence of the Schwarz method. We are currently studying a way to introduce a much better parameter for the convergence of block Jacobi, without changing however the DG approximation properties.

Acknowledgements We would like to thank BLANCA AYUSO for her useful comments.

References

1. Douglas N. Arnold, Franco Brezzi, et al. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM J. Num. Anal.*, 39(5):1749–1779, 2002.
2. Bernardo Cockburn, Jayadeep Gopalakrishnan, and Raytcho Lazarov. Unified hybridization of discontinuous galerkin, mixed, and continuous galerkin methods for second order elliptic problems. *SIAM J. Num. Anal.*, 47(2):1319–1365, 2009.
3. Paul Castillo, Bernardo Cockburn, et al. An a Priori Error Analysis of the Local Discontinuous Galerkin Method for Elliptic Problems. *SIAM J. Num. Anal.*, 38(5):1676–1706, 2001.
4. M. J. Gander. Schwarz methods over the course of time. *Electronic Transactions on Numerical Analysis*, 31:228–255, 2008.
5. M. J. Gander. Optimized Schwarz Methods. *SIAM J. Num. Anal.*, 44(2):699-731, 2006.
6. S. H. Lui. A lions non-overlapping domain decomposition method for domains with an arbitrary interface. *IMA Journal of Numerical Analysis*, 29(2):332–349, 2009.

Overlapping domain decomposition methods with `FreeFem++`

Pierre Jolivet[1,3], Frédéric Hecht[1], Frédéric Nataf[1], and Christophe Prud'homme[2]

1 Introduction

Developping an efficient and versatile framework for finite elements domain decomposition methods can be a hard task because of the mathematical genericity of finite element spaces, the complexity of handling arbitrary meshes and so on. The purpose of this note is to present one way to implement such a framework in the context of overlapping decompositions. In section 2, the basics for one-level overlapping methods is introduced, in section 3, a second level is added to the original framework to ensure scalability using a portable `C++` library, and section 4 gathers some numerical results. `FreeFem++` will be used for the computations of finite element matrices, right hand side and mesh generation, but the work here is also applicable to other Domain-Specific (Embedded) Language such as `deal.II` [3], `Feel++` [12], `GetFem++`...

2 One-level methods

Let $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) be a domain whose associated mesh can be partitioned into N non-overlapping meshes $\{\mathcal{T}_i\}_{1\leqslant i\leqslant N}$ using graph partitioners such as METIS [10] or SCOTCH [5]. Let V be the finite element space spanned by the finite set of n basis functions $\{\phi_i\}_{1\leqslant i\leqslant n}$ defined on Ω, and $\{V_i\}_{1\leqslant i\leqslant N}$ be the local finite element spaces defined on the domains associated to each $\{\mathcal{T}_i\}_{1\leqslant i\leqslant N}$. Typical finite element discretizations of a symmetric, coercive bilinear form $a : V \times V \to \mathbb{R}$ yield the following system to solve :

$$Ax = b, \tag{1}$$

where $(A_{ij})_{1\leqslant i,j\leqslant n} = a(\phi_j, \phi_i)$, and $(b_i)_{1\leqslant i\leqslant n} = (f, \phi_i)$, f being in the dual space V^*. Let an integer δ be the level of overlap: $\left\{\mathcal{T}_i^\delta\right\}_{1\leqslant i\leqslant N}$ is an overlapping decomposition and if we consider the restrictions $\{R_i\}_{1\leqslant i\leqslant N}$ from V to $\left\{V_i^\delta\right\}_{1\leqslant i\leqslant N}$, the local finite element spaces on $\left\{\mathcal{T}_i^\delta\right\}_{1\leqslant i\leqslant N}$, and a local partition of unity $\{D_i\}_{1\leqslant i\leqslant N}$ such that

[1] Laboratoire Jacques-Louis Lions, CNRS UMR 7598, Université Pierre et Marie Curie, 75005 Paris, France, e-mail: `{jolivet}{hecht}{nataf}@ann.jussieu.fr` · [2] Institut de Recherche Mathématique Avancée, CNRS UMR 7501, Université de Strasbourg, 7 rue René Descartes, 67084 Strasbourg Cedex, France, e-mail: `prudhomme@unistra.fr` · [3] Laboratoire Jean Kuntzmann, CNRS UMR 5224, Université Joseph Fourier, 51 rue des Mathématiques, BP53, 38041 Grenoble Cedex 9, France, e-mail: `jolivet@imag.fr`

$$\sum_{j=1}^{N} R_j^T D_j R_j = I. \tag{2}$$

Then a common one-level preconditioner for system (1) introducted in [4] is

$$\mathscr{P}_{\mathrm{RAS}}^{-1} = \sum_{i=1}^{N} R_i^T D_i (R_i A R_i^T)^{-1} R_i. \tag{3}$$

The global matrix A is never assembled, instead, we build locally $A_i^{\delta+1}$ the stiffness matrix yielded by the discretization of a on $V_i^{\delta+1}$, and we remove the columns and rows associated to degrees of freedom lying on elements of $\mathscr{T}_i^{\delta+1} \setminus \mathscr{T}_i^{\delta}$, this yields $A_i = R_i A R_i^T$. The distributed sparse matrix-vector product Ax for $x \in \mathbb{R}^n$ can be computed using point-to-point communications and the partition of unity without having to store the global *distributed* matrix A. Indeed, using (2), if one looks at the local components of Ax, that is $R_i Ax$, then one can write, introducing $\mathscr{O}_i$ the set of neighboring subdomains to i, i.e. $\left\{ j \,:\, \mathscr{T}_i^{\delta} \cap \mathscr{T}_j^{\delta} \neq \emptyset \right\}$:

$$\begin{aligned} R_i Ax &= \sum_{j=1}^{N} R_i A R_j^T D_j R_j x && (4)\\ &= A_i D_i R_i x + \sum_{j \in \mathscr{O}_i} R_i R_j^T A_j D_j R_j x. && (5) \end{aligned}$$

since it can be checked that

$$\forall x \in \mathbb{R}^n,\ R_i A R_j^T D_j R_j x = R_i R_j^T R_j A R_j^T D_j R_j x \tag{6}$$

The sparse matrix-sparse matrix products $R_i R_j^T$ are nothing else than point-to-point communications from neighbors j to i.
In `FreeFem++`, stiffness matrices such as $A_i^{\delta+1}$ and right-hand sides are assembled as follows (a simple 2D Laplacian is considered here):

```
mesh Th;                  // Th is a local 2D mesh (for example T_i^{δ+1})
fespace Vh(Th, Pk);       // Vh is a local finite element space
varf a(u, v) = int2d(dx(u) * dx(v) + dy(u) * dy(v))
             + int2d(f * v) + BC;
matrix A = a(Vh, Vh);     // A is a sparse matrix stored in the CSR format
Vh rhs;                   // rhs is a function lying in the FE space Vh
rhs[] = a(0, Vh);         // Its values are set to solve Ax = rhs
```

The mesh `Th` can either be created on the fly by `FreeFem++`, or it can be loaded from a file generated offline by Gmsh [6], for example when dealing with complex geometries. By default, `FreeFem++` handles continuous piecewise linear, quadratic, cubic, quartic finite elements, and other traditionnal FE like Raviart-Thomas 1, Morley, ... The boundary conditions depend on the label set on the mesh. For example, if one wants to impose penalized homogeneous Dirichlet boundary conditions on the label `1` of the boundary of `Th`, then one just has to add

`+ on(1, u = 0)` in the definition of the `varf`. For a more detailed introduction to FreeFem++ with abundant examples, interested readers should visit `http://www.freefem.org/ff++` or see [9].
The partition of unity D_i is built using a continuous piecewise linear approximation of

$$\chi_i = \frac{\widetilde{\chi}_i}{\widetilde{\chi}_i + \sum_{j \in \mathscr{O}_i} \widetilde{\chi}_j|_{\mathscr{T}_i^\delta \cap \mathscr{T}_j^\delta}}, \tag{7}$$

where $\widetilde{\chi}_i$ is defined as

$$\widetilde{\chi}_i = \begin{cases} 1 & \text{on all vertices of } \mathscr{T}_i \\ 1 - \dfrac{m}{\delta} & \text{on all vertices of } \mathscr{T}_i^m \setminus \mathscr{T}_i^{m-1} \; \forall m \in [1;\delta]\,. \end{cases} \tag{8}$$

3 Two-level methods

It is well known that one-level domain decomposition methods as depicted in section 2 do suffer from poor conditioning when used with many subdomains, [16]. In this section, we present a new C++ library, independent of the finite element backend used, that assembles efficiently a coarse operator that will be used in section 4 to ensure scalability of our framework. The theoretical foundations for the construction of the coarse operator are presented in [14]. From a practical point of view, after building each local solver A_i, three dependent operators are needed :

(i) the deflation matrix Z,
(ii) the coarse operator $E = Z^T A Z$,
(iii) the actual precontioner $\mathscr{P}^{-1}_{\text{A-DEF1}} = \mathscr{P}^{-1}_{\text{RAS}}(I - AZE^{-1}Z^T) + ZE^{-1}Z^T$, thoroughly studied in [15].

In [14], the deflation matrix is defined as :

$$Z = \left[R_1^T W_1 \; R_2^T W_2 \; \cdots \; R_N^T W_N\right] \in \mathbb{R}^n \times \mathbb{R}^{\sum_{i=1}^N \nu_i} \tag{9}$$

where

$$\left\{W_i = \left[D_i \Lambda_{i_1} \; D_i \Lambda_{i_2} \; \cdots \; D_i \Lambda_{i_{\nu_i}}\right] \in \mathbb{R}^{n_i} \times \mathbb{R}^{\nu_i}\right\}_{1 \leqslant i \leqslant N} \tag{10}$$

ν_i is a threshold criterion used to select the eigenvectors Λ_i associated to the smallest eigenvalues in magnitude of the following *local* generalized eigenvalue problem:

$$A_i^\delta \Lambda_i = \lambda_i D_i R_{i,0}^T R_{i,0} A_i^\delta D_i \Lambda_i$$

where A_i^δ is the matrix yielded by the discretization of a on V_i^δ, and $R_{i,0}$ is the restriction operator from $\mathscr{T}_i^\delta$ to the overlap $\mathscr{T}_i^\delta \cap \left(\cup_{j \in \mathscr{O}_i} \mathscr{T}_j^\delta\right)$. In FreeFem++, sparse eigenvalue problems are solved either with SLEPc [8] or ARPACK [11]. The latter seems to yield better performance in our simulations. Given, for each MPI process,

the local matrix A_i, the local partition of unity D_i, the set of eigenvalues $\left\{\Lambda_{i_j}\right\}_{1\leqslant j\leqslant \nu_i}$ and the set of neighboring subdomains $\mathcal{O}_i$, our library assembles E without having to assemble A and to store Z, and computes its LU or LDL^T factorization using either MUMPS [1, 2], PARDISO [13] or PaStiX [7]. Moreover, all linear algebra related computations (e.g. sparse matrix-vector products) within our library are performed using Intel MKL, or can use user-supplied functions, for example those from within the finite element Domain-Specific (Embedded) Language. Assembling E is done in two steps: local computations and then renumbering.

- first, compute *local* vector-sparse matrix-vector triple products which will be used to assemble the diagonal blocks of E. For a given row in E, off-diagonal values are computed using *local* sparse matrix-vector products coupled with point-to-point communications with the neighboring subdomains: the sparsity pattern of the coarse operator is similar to the dual graph of the mesh partitioning (hence it is denser in 3D than in 2D),
- then, renumber the *local* entries computed previously in the *distributed* matrix E.

Only few processes are in charge of renumbering entries into E. Those processes will be refered to in the rest of this note as *master processes*. Any non master process has to send the rows it has previously computed to a specific master process. The master processes are then able to place the entries received at the right row and column indices. To allow an easy incremental matrix construction, E is assembled using the COO format. If need be, it is converted afterwards to the CSR format. Note here that MUMPS only supports the COO format while PARDISO and PaStiX work with the CSR format.
After renumbering, the master processes are also the one in charge of computing the factorization of the coarse operator. The number of master processes is a runtime constant, and our library is in charge of creating the corresponding MPI communicators. Even with "large" coarse operators of sizes of around $100\,000 \times 100\,000$, less than few tens of master processes usually perform the job quite well: computing all entries, renumbering and performing numerical factorization take around 15 seconds when dealing with thousands of slave processes.
A routine is then callable to solve the equation $Ex = y$ for an arbitrary $y \in \mathbb{R}^{\Sigma_{i=1}^{N} \nu_i}$, which in our case is used at each iteration of our Krylov method preconditioned by $\mathscr{P}^{-1}_{\text{A-DEF1}}$. Once again, the deflation matrix Z is not stored as the products $Z^T x \in \mathbb{R}^{\Sigma_{i=1}^{N} \nu_i}$ and $Zy \in \mathbb{R}^n$ can be computed explicitely with a *global* matrix-free method (we only use the *local* W_i plus point-to-point communications with neighboring subdomains).

4 Numerical results

Results in this section were obtained on Curie, a Tier-0 system for PRACE composed of 5040 nodes made of 2 eight-core Intel Sandy Bridge processors clocked

at 2.7 GHz. The interconnect is an InfiniBand QDR full fat tree network. We want here to assess the capability of our framework to scale:

(i) strongly: for a given *global* mesh, the number of subdomains increases while *local* mesh sizes are kept constant (i.e. local problems get smaller and smaller),
(ii) weakly: for a given *global* mesh, the number of subdomains increases while *local* mesh sizes are refined (i.e. local problems have a constant size).

We don't time the generation of the mesh and partition of unity. Assembly and factorization of the local stiffness matrices, resolution of the generalized eigenvalue problems, construction of the coarse operator and time elapsed for the convergence of the Krylov method are the important procedures here. The Krylov method used is the GMRES, it is stopped when the relative residual error is inferior to $\varepsilon = 10^{-6}$ in 2D, and 10^{-8} in 3D. All the following results where obtained using a LDL^T factorization of the local solvers A_i^δ and the coarse operator E using MUMPS (with a MPI communicator set to respectively `MPI_COMM_SELF` or the communicator created by our library binding master processes).

First, the system of linear elasticity with highly heterogeneous elastic moduli is solved with a minimal geometric overlap of one mesh element. Its variational formulation reads:

$$\int_\Omega \lambda \nabla \cdot u \nabla \cdot v + 2\mu\varepsilon(u)^T\varepsilon(v) + \int_\Omega f \cdot v + \int_{\partial\Omega} g \cdot v \tag{11}$$

where

- λ and μ are the Lamé parameters such that $\mu = \dfrac{E}{2(1+\nu)}$ and $\lambda = \dfrac{E\nu}{(1+\nu)(1-2\nu)}$ (E being Young's modulus and ν Poisson's ratio). They are chosen to vary between two sets of values, $(E_1, \nu_1) = (2\cdot 10^{11}, 0.25)$, and $(E_2, \nu_2) = (10^8, 0.4)$.
- ε is the linearized strain tensor and f the volumetric forces (here, we just consider gravity).

Because of the overlap and the duplication of unkowns, increasing the number of subdomains means that the number of unknowns increases also slightly, even though the number of mesh elements (triangles or tetrahedra in the case of `FreeFem++`) is the same. In 2D, we use piecewise cubic basis functions on an unstructured *global* mesh made of 110 million elements, and in 3D, piecewise quadratic basis functions on an unstructured *global* mesh made of 20 million elements. This yields a symmetric system of roughly 1 billion unkowns in 2D and 80 million unknowns in 3D. The geometry is a simple $[0;1]^d \times [0;10]$ beam ($d = 1$ or 2) partitioned with METIS.

Solving the 2D problem initially on 1 024 processes takes 227 seconds, on 8192 processes, it takes 31 seconds (quasioptimal speedup). With that many subdomains, the coarse operator E is of size $121\,935 \times 121\,935$. It is assembled and factorized in 7 seconds by 12 master processes. For the 3D problem, it takes initially 373 seconds. At peak performance, near 6 144 processes, it takes 35 seconds (superoptimal speedup). This time, the coarse operator is of size $92\,160 \times 92\,160$ and is assembled and factorized by 16 master processes in 11 seconds.

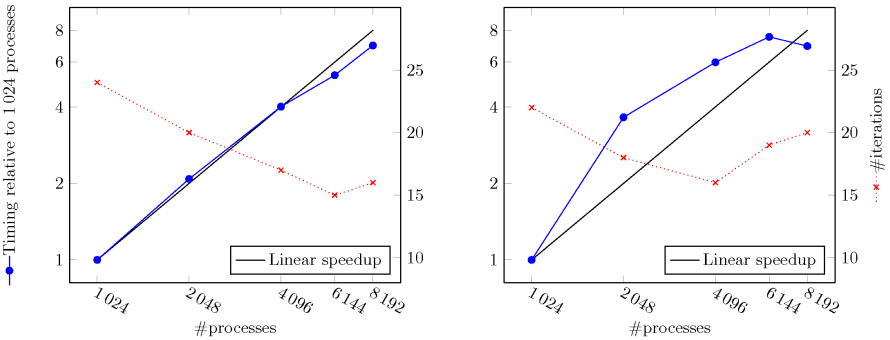


Fig. 1 Linear elasticity test cases. 2D on the left, 3D on the right. Strong scaling

Moving on to the weak scaling propreties of our framework, the problem we now solve is a scalar equation of diffusivity with highly heterogeneous coefficients (varying from 1 to 10^5) on $[0;1]^d$ ($d = 2$ or 3). Its variational formulation reads:

$$\int_\Omega \kappa \nabla u \cdot \nabla v + \int_\Omega f \cdot v \tag{12}$$

The targeted number of unkowns per subdomains is kept constant at approximately 800 thousands in 2D, and 120 thousands in 3D (once again with $\mathbb{P}_3$ and $\mathbb{P}_2$ finite elements respectively).

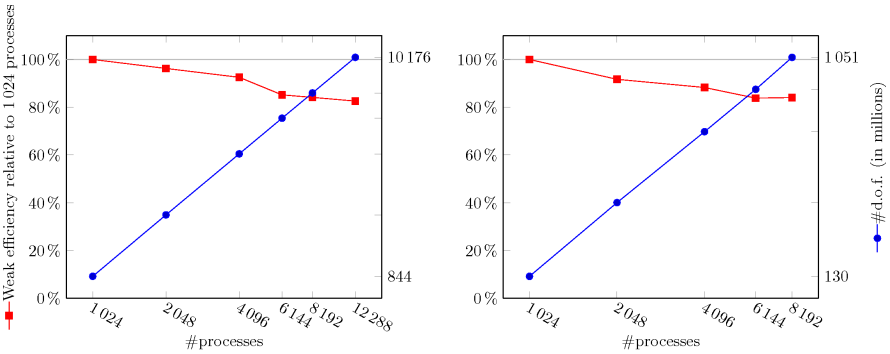


Fig. 2 Diffusion equation test cases. 2D on the left, 3D on the right. Weak scaling

In 2D, the initial extended system (with the duplication of unkowns) is made of 800 million unkowns and is solved in 141 seconds. Scaling up to 12 288 processes yields a system of 10 billion unkowns solved in 172 seconds, hence an efficiency of $\frac{141}{172} \approx 82\%$. In 3D, the initial system is made of 130 million unkowns and is solved in 127 seconds. Scaling up to 8192 processes yields a system of 1 billion unkowns solved in 152 seconds, hence an efficiency of $\frac{127}{152} \approx 83\%$.

5 Conclusion

This note clearly shows that our framework scales on very large architectures for solving linear positive definite systems using overlapping decompositions with many subdomains. It is currently being extended to support nonlinear problems (namely in the field of nonlinear elasticity) and we should be able to provide similar functionalities for non-overlapping decompositions. It should be noted that the heavy use of threaded (sparse) BLAS and LAPACK routines (via Intel MKL, PARDISO, and the Reverse Communication Interface of ARPACK) has already helped us to get a quick glance at how the framework performs using hybrid parallelism. We are confident that using this novel paradigm, we can still improve our scaling results in the near future by switching the value of OMP_NUM_THREADS to a value greater than 1.

Acknowledgements This work has been supported in part by ANR through COSINUS program (project PETALh no. ANR-10-COSI-0013 and projet HAMM no. ANR-10-COSI-0009). It was granted access to the HPC resources of TGCC@CEA made available within the Distributed European Computing Initiative by the PRACE-2IP, receiving funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement RI-283493.

References

1. Amestoy, P., Duff, I., L'Excellent, J.Y., Koster, J.: A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM Journal on Matrix Analysis and Applications **23**(1), 15–41 (2001)
2. Amestoy, P., Guermouche, A., L'Excellent, J.Y., Pralet, S.: Hybrid scheduling for the parallel solution of linear systems. Parallel computing **32**(2), 136–156 (2006)
3. Bangerth, W., Hartmann, R., Kanschat, G.: deal.II — a general-purpose object-oriented finite element library. ACM Transactions on Mathematical Software **33**(4), 24–27 (2007)
4. Cai, X.C., Sarkis, M.: Restricted additive Schwarz preconditioner for general sparse linear systems. SIAM Journal on Scientific Computing **21**(2), 792–797 (1999)
5. Chevalier, C., Pellegrini, F.: PT-Scotch: a tool for efficient parallel graph ordering. Parallel Computing **34**(6), 318–331 (2008)
6. Geuzaine, C., Remacle, J.F.: Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering **79**(11), 1309–1331 (2009)
7. Hénon, P., Ramet, P., Roman, J.: PaStiX: a high performance parallel direct solver for sparse symmetric positive definite systems. Parallel Computing **28**(2), 301–321 (2002)
8. Hernandez, V., Roman, J., Vidal, V.: SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. ACM Transactions on Mathematical Software **31**(3), 351–362 (2005)
9. Jolivet, P., Dolean, V., Hecht, F., Nataf, F., Prud'homme, C., Spillane, N.: High performance domain decomposition methods on massively parallel architectures with FreeFem++. Journal of Numerical Mathematics **20**(4), 287–302 (2012)
10. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing **20**(1), 359–392 (1998)
11. Lehoucq, R., Sorensen, D., Yang, C.: ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods, vol. 6. Society for Industrial and Applied Mathematics (1998)

12. Prud'homme, C., Chabannes, V., Doyeux, V., Ismail, M., Samake, A., Pena, G.: Feel++: A computational framework for Galerkin methods and advanced numerical methods. In: ESAIM: Proceedings, vol. 38, pp. 429–455 (2012)
13. Schenk, O., Gärtner, K.: Solving unsymmetric sparse systems of linear equations with PARDISO. Future Generation Computer Systems **20**(3), 475–487 (2004)
14. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R.: A robust two-level domain decomposition preconditioner for systems of PDEs. Comptes Rendus Mathematique **349**(23), 1255–1259 (2011)
15. Tang, J., Nabben, R., Vuik, C., Erlangga, Y.: Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. Journal of Scientific Computing **39**(3), 340–370 (2009)
16. Toselli, A., Widlund, O.: Domain decomposition methods — algorithms and theory, *Series in Computational Mathematics*, vol. 34. Springer (2005)

On the influence of curvature on transmission conditions

Hélène Barucq[1], Martin J. Gander[2], and Yingxiang Xu[3]

1 Introduction

Domain decomposition methods are both highly successful parallel solvers and also important modeling tools, since problems in subdomains can be treated by adapted methods to the physics in each subdomain. Subdomain boundaries are therefore rarely straight lines. The focus of this paper is to study the influence of curvature on transmission conditions used in optimized Schwarz methods. For straight interfaces and simple geometries, optimized interface conditions are typically determined using Fourier analysis, see for example [4] and references therein. Asymptotically, these optimized conditions are still valid for curved interfaces, as shown in [5, 6]. Since however the curvature is the most important information for a smooth curve, we want to study in this paper if and how the interface curvature influences the constants in the optimized parameters.

We consider the model problem

$$(\Delta - \eta)u = f, \quad \text{on } \Omega = \mathbb{R}^2,\ \eta > 0, \tag{1}$$

and we require the solution to decay at infinity. As shown in Fig. 1 on the left, we decompose Ω into two overlapping subdomains $\Omega_1 = (-\infty, a(y)) \times \mathbb{R}$ and $\Omega_2 = (b(y), \infty) \times \mathbb{R}$, where Γ_1 given by $a(y)$ and Γ_2 given by $b(y)$ are smooth curves satisfying $a(y) \geq b(y)$. A general parallel Schwarz algorithm is then given by

$$\begin{aligned} (\Delta - \eta)u_i^n &= f && \text{in } \Omega_i, \\ \mathscr{B}_i(u_i^n) &= \mathscr{B}_i(u_j^{n-1}) && \text{on } \Gamma_i,\ 1 \leq i \neq j \leq 2, \end{aligned} \tag{2}$$

where $\mathscr{B}_i, i = 1,2$, are transmission conditions to be chosen. If $\mathscr{B}_i$, $i = 1,2$ are chosen as $\partial_{n_i} + DtN_i$, with DtN_i the Dirichlet to Neumann operators, the iterates will converge in two steps [4]. These operators are however non-local, and thus difficult to use in practice. Therefore, local approximations are used in optimized Schwarz methods. We study in what follows such local approximations, obtained by microlocal analysis, and by studying a circular model problem, with the goal to investigate how the curvature influences these approximations.

[1] MAGIQUE-3D (INRIA Bordeaux - Sud-Ouest) INRIA-CNRS-Université de Pau et des Pays de l'Adour, BP 1155, F64013 PAU, France, e-mail: helene.barucq@inria.fr ·[2] Section de Mathématiques, Université de Genève, 2-4 rue du Lièvre, CP 64, 1211 Genève 4, Suisse, e-mail: Martin.Gander@unige.ch ·[3] School of Mathematics and Statistics, Northeast Normal University, Changchun 130024, China, e-mail: yxxu@nenu.edu.cn, partly supported by NSFC-11201061 and CPSF-2012M520657.

2 Transmission conditions based on micro-local analysis

Micro-local analysis is a well established technique for the design and study of absorbing boundary conditions, where it is used to approximate the *DtN*, see [2] and references therein. We use in this section micro-local analysis to develop and analyze transmission conditions. As in [2], we consider local coordinates composed by the curvilinear abscissa s and the variable r along the normal direction. In these local coordinates, the model problem (1) can be rewritten as

$$\mathscr{L}u := \partial_{rr}u + \frac{\kappa}{h}\partial_r u + \frac{1}{h}\partial_s(\frac{\partial_s u}{h}) - \eta u = f, \tag{3}$$

where $\kappa = \kappa(s)$ is the curvature of the curve Γ_i at the parameter s, and $h = h(r,s) = 1 + r\kappa(s)$. The symbol of the operator $\mathscr{L}$ is given by

$$\hat{\mathscr{L}} = \partial_{rr} + \frac{\kappa}{h}\partial_r + \frac{i}{h}\partial_s(\frac{1}{h})\xi - \frac{1}{h^2}\xi^2 - \eta. \tag{4}$$

A pseudodifferential operator $\mathscr{P}$ is defined by $\mathscr{P}u(x) := \int e^{ix\cdot\xi} p(x,\xi)\hat{u}(\xi)d\xi$, provided its symbol $p(x,\xi) \in S^m$, i.e. for every compact set K in $\mathbb{R}^n$ and for every α,β there exists $c = c(\alpha,\beta,K)$ s.t. for all $(x,\xi) \in K \times \mathbb{R}^n$, $|\partial_\xi^\alpha D_x^\beta p(x,\xi)| \leq c(1+|\xi|)^{m-|\alpha|}$. Based on the Nirenberg's factorization theorem, there exist two classical pseudo-differential operators Λ^- and Λ^+ of order $+1$, depending smoothly on r, such that

$$\mathscr{L}u = (\partial_r + \Lambda^-)(\partial_r + \Lambda^+)u, \tag{5}$$

which can be expanded as

$$\mathscr{L}u = \partial_{rr}u + (\Lambda^- + \Lambda^+)\partial_r u + \mathrm{op}(\partial_r\lambda^+)u + \Lambda^-\Lambda^+ u, \tag{6}$$

where $\mathrm{op}(\partial_r\lambda^+)$ is the operator whose symbol is $\partial_r\lambda^+$. In (5) and (6), the symbol '$=$' must be interpreted as equal up to a C^∞-regularizing operator, since the symbols of Λ^+ and Λ^- are explicitely defined by the factorization process up to a symbol in

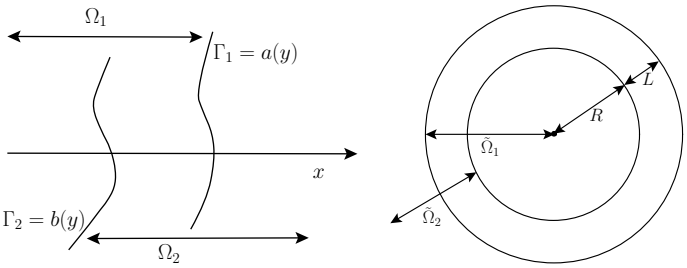


Fig. 1 An arbitrary domain decomposition with curved interfaces (left) and a circular domain decomposition (right)

$S^{-\infty}$. Identifying (3) and (6) we get

$$\Lambda^- + \Lambda^+ = \frac{\kappa}{h}, \quad \mathrm{op}(\partial_r \lambda^+) + \Lambda^- \Lambda^+ = \frac{1}{h}\partial_s(\frac{\partial_s}{h}) - \eta. \tag{7}$$

Due to the integral representation formula of pseudo-differential operators, the operators Λ^- and Λ^+ are determined by their symbols. Using the calculus of pseudo-differential operators, system (7) can be written at the symbol level,

$$\lambda^- + \lambda^+ = \frac{\kappa}{h}, \quad \sum_{\alpha=0}^{+\infty} \frac{(-i)^\alpha}{\alpha!} \partial_\xi^\alpha \lambda^- \partial_s^\alpha \lambda^+ + \partial_r \lambda^+ = -\eta - h^{-2}\xi^2 + \frac{i}{h}\partial_s(\frac{1}{h})\xi, \tag{8}$$

where $\lambda^\pm \sim \sum_{j=-1}^{+\infty} \lambda_{-j}^\pm$ are the symbols of $\Lambda^\pm$. The goal is now to determine the symbols λ^- and λ^+: from the first equation in (8), we get

$$\lambda_{-j}^- + \lambda_{-j}^+ = 0, \text{ if } j \neq 0 \text{ and } \lambda_0^- + \lambda_0^+ = \frac{\kappa}{h}. \tag{9}$$

By identifying the homogeneous symbols of highest degree, we obtain

$$\lambda_1^- \lambda_1^+ = -h^{-2}\xi^2 - \eta, \tag{10}$$

where η is considered to be an operator of order 2, see Section 3 of [2] for details. Therefore, we have

$$\lambda_1^+ = \sqrt{h^{-2}\xi^2 + \eta} \quad \text{and} \quad \lambda_1^- = -\sqrt{h^{-2}\xi^2 + \eta}. \tag{11}$$

Going further with the identification of the homogeneous symbols of the next higher degree, we find a relation between the unknowns λ_0^- and λ_0^+,

$$\lambda_1^- \lambda_0^+ + \lambda_0^- \lambda_1^+ - i\partial_\xi \lambda_1^- \partial_s \lambda_1^+ + \partial_r \lambda_1^+ = \frac{i}{h}\partial_s(\frac{1}{h})\xi. \tag{12}$$

Eliminating λ_1^- and λ_0^-, we get

$$\lambda_0^+ = \frac{1}{2\lambda_1^+}(\frac{\kappa}{h}\lambda_1^+ + i\partial_\xi \lambda_1^+ \partial_s \lambda_1^+ + \partial_r \lambda_1^+ - \frac{i}{h}\partial_s(\frac{1}{h})\xi). \tag{13}$$

We can derive a recursive formula from similar relations for lower degrees of homogeneity. First, we rewrite the left-hand side of the second equation in (8) as

$$\sum_{\alpha=0}^{+\infty} \frac{(-i)^\alpha}{\alpha!} \sum_{j=-1}^{+\infty} \partial_\xi^\alpha \lambda_{-j}^- \sum_{k=-1}^{+\infty} \partial_s^\alpha \lambda_{-k}^+ + \sum_{l=-1}^{+\infty} \partial_r \lambda_{-l}^+. \tag{14}$$

Since $\partial_\xi^\alpha \lambda_{-j}^- \partial_s^\alpha \lambda_{-k}^+ \in S^{-(j+k+\alpha)}$, the homogeneous part of degree $-m$ in (14) for any non-negative integer m is

$$\sum_{\alpha=0}^{m+2}\frac{(-i)^\alpha}{\alpha!}\sum_{\substack{j+k=m-\alpha,\\ j\geq -1,k\geq -1}}\partial_\xi^\alpha\lambda_{-j}^-\partial_s^\alpha\lambda_{-k}^+ + \partial_r\lambda_{-m}^+.$$

Identifying symbols of the same homogeneity in (8) leads to

$$\sum_{\alpha=0}^{m+2}\frac{(-i)^\alpha}{\alpha!}\sum_{\substack{j+k=m-\alpha,\\ j\geq -1,k\geq -1}}\partial_\xi^\alpha\lambda_{-j}^-\partial_s^\alpha\lambda_{-k}^+ + \partial_r\lambda_{-m}^+ = 0.$$

Using that $\lambda_{-m-1}^- = -\lambda_{-m-1}^+$, from the previous equation, the symbol λ_{-m-1}^+ for $m\geq 0$ can be recursively expressed from homogeneous symbols of higher order by

$$\lambda_{-m-1}^+ = \frac{1}{2\lambda_1^+}\Big(\sum_{\substack{j+k=m,\\ j\geq 0,k\geq 0}}\lambda_{-j}^-\lambda_{-k}^+ + \sum_{\alpha=1}^{m+2}\frac{(-i)^\alpha}{\alpha!}\sum_{\substack{j+k=m-\alpha,\\ j\geq -1,k\geq -1}}\partial_\xi^\alpha\lambda_{-j}^-\partial_s^\alpha\lambda_{-k}^+ + \partial_r\lambda_{-m}^+\Big). \tag{15}$$

Let ℓ be a positive integer, and μ be the symbol of the pseudo-differential operator $\mathrm{op}(\mu)$ defined on $\Gamma_i\times(-\delta,\delta)$, $i=1,2$, such that $\sum_{-1\leq j\leq p}\lambda_{-j}^+ - \mu$ is of order $(1/\sqrt{\eta})^{\ell-1}$ for all sufficiently large p. Denoting by $\tilde{\mu}$ the symbol defined on $\Gamma_i, i=1,2$ by $\tilde{\mu} := \mu|_{r=0}$, and choosing as transmission condition $\mathscr{B}_i = \partial_{n_i} + \mathrm{op}(\tilde{\mu})$ on Γ_i, we obtain the MATCs (Micro-local Analysis based Transmission Conditions) of order $\ell/2$ as

$$\mathscr{B}_i = \partial_{n_i} + \mathrm{op}\Big(\sum_{-1\leq j\leq \ell-2}\lambda_{-j}^+\Big), \quad \text{on } \Gamma_i,\ i=1,2. \tag{16}$$

From (15), note that λ_{-m-1}^+ still contains the term $\lambda_1^+ = \sqrt{h^{-2}\xi^2+\eta}$, and thus results in non-local transmission conditions. To obtain local transmission conditions, we use a Taylor expansion in ξ of the symbols λ_{-j}^+, $-1\leq j\leq 2$ to the order shown as index in the parentheses below, and obtain the following MATCs:

MATC1 $\quad \mathscr{B}_i(u) = \partial_{n_i}u + \mathrm{op}((\lambda_1^+)_0)u = \partial_{n_i}u + \sqrt{\eta}u;$

MATC2 $\quad \mathscr{B}_i(u) = \partial_{n_i}u + \mathrm{op}((\lambda_1^+)_0 + (\lambda_0^+)_0)u = \partial_{n_i}u + (\sqrt{\eta}+\frac{\kappa}{2})u;$

MATC3 $\quad \mathscr{B}_i(u) = \partial_{n_i}u + \mathrm{op}(\sum_{j=-1}^{2}(\lambda_{-j}^+)_0)u = \partial_{n_i}u + (\sqrt{\eta}+\frac{\kappa}{2} - \frac{\kappa^2}{8\sqrt{\eta}} + \frac{\kappa^3+\frac{d^2}{ds^2}\kappa(s)}{8\eta})u;$

MATC4 $\quad \mathscr{B}_i(u) = \partial_{n_i}u + \mathrm{op}(\sum_{j=-1}^{1}(\lambda_{-j}^+)_1)u = \partial_{n_i}u + (\sqrt{\eta}+\frac{\kappa}{2} - \frac{1}{8}\frac{\kappa^2}{\sqrt{\eta}})u - \frac{\frac{d}{ds}\kappa(s)}{2\eta}\partial_s u;$

MATC5 $\quad \mathscr{B}_i(u) = \partial_{n_i}u + \mathrm{op}((\lambda_1^+)_2)u = \partial_{n_i}u + \sqrt{\eta}u - \frac{1}{2\sqrt{\eta}}\partial_s^2 u;$

MATC6
$$\begin{aligned}\mathscr{B}_i(u) = \partial_{n_i}u + \mathrm{op}(\textstyle\sum_{j=-1}^{2}(\lambda_{-j}^+)_{-2})u &= \partial_{n_i}u + (\sqrt{\eta}+\tfrac{\kappa}{2} - \tfrac{1}{8}\tfrac{\kappa^2}{\sqrt{\eta}} + \tfrac{1}{8}\tfrac{\kappa^3+\frac{d^2}{ds^2}\kappa(s)}{\eta})u\\ &+ (\tfrac{\frac{d}{ds}\kappa(s)}{2\eta} - \tfrac{13}{8}\tfrac{\kappa(s)\frac{d}{ds}\kappa(s)}{\eta^{\frac{3}{2}}})\partial_s u - (\tfrac{1}{2\sqrt{\eta}} - \tfrac{1}{2}\tfrac{\kappa}{\eta} + \tfrac{13}{16}\tfrac{\kappa^2}{\eta^{\frac{3}{2}}} - \tfrac{7}{8}\tfrac{2\kappa^3+\frac{d^2}{dx^2}\kappa(s)}{\eta^2})\partial_s^2 u,\end{aligned}$$

where the MATC1–3 are of order 0, MATC4 is of order 1, and MATC5 and MATC6 are of order 2. Note how the curvature $\kappa(s)$ enters these transmission conditions.

3 Transmission conditions based on a circular model problem

For optimized Schwarz methods, transmission conditions are often analyzed and optimized for a model problem, see [4]. Following this principle, we consider a circular decomposition of the domain $\Omega = \mathbb{R}^2$ as shown in Fig. 1 on the right,

$$\tilde{\Omega}_1 = \{(x,y)|\sqrt{x^2+y^2} < R_1 = R+L\}, \quad \tilde{\Omega}_2 = \{(x,y)|R_2 = R < \sqrt{x^2+y^2} < \infty\}.$$

In this setting, the curvature of the interface enters naturally, $\kappa(s) = 1/R$. Using polar coordinates, a general Schwarz algorithm for this decomposition is

$$\begin{aligned} \partial_{rr}u_i^n + \tfrac{1}{r}\partial_r u_i^n + \tfrac{1}{r^2}\partial_{\theta\theta}u_i^n - \eta u_i^n &= f && \text{in } \tilde{\Omega}_i, \\ \mathscr{B}_i(u_i^n) &= \mathscr{B}_i(u_j^{n-1}) && \text{on } r = R_i,\ 1 \le i \ne j \le 2. \end{aligned} \tag{17}$$

In the classical Schwarz algorithm, one uses for $\mathscr{B}_i$ the identity operator in (17). Using Fourier series in the angular variable, we obtain after a short calculation for the convergence factor ρ_{cla} in this case (for details of such calculations, see [3])

$$\rho_{cla} = \rho_{cla}(k,R,L,\eta) := \frac{I_k(\sqrt{\eta}R)}{K_k(\sqrt{\eta}R)}\frac{K_k(\sqrt{\eta}(R+L))}{I_k(\sqrt{\eta}(R+L))}, \quad \forall k \in \mathbb{R}, \tag{18}$$

where $I_k(\cdot)$ and $K_k(\cdot)$ are the modified Bessel functions of the first (exponentially increasing) and the second kind (exponentially decreasing), see [1]. Hence, for an overlap $L > 0$, the classical Schwarz algorithm converges, with the asymptotic estimate

$$\sup_{k_{\min} \le k \le k_{\max}} \rho_{cla} = 1 - G_{min}L + O(L^2), \quad G_{min} = \frac{1}{RI_{k_{min}}(\sqrt{\eta}R)K_{k_{min}}(\sqrt{\eta}R)},$$

where $k_{\min}$ and k_{max} denote the estimates of the lowest and highest relevant numerical frequencies respectively. If there is no overlap, the method does not converge.

Optimized Schwarz methods are based on linear operators S_i, $i = 1,2$ along the interface, here in the θ direction, with symbols σ_i, and $\mathscr{B}_i(u) = \partial_r u - S_i u$ in (17). This results in methods with convergence factors $\rho_{opt}(k,L,R,\eta,\sigma_1,\sigma_2)$ given by (for details, see [3])

$$\rho_{opt} = \frac{\frac{\partial_r K_k(\sqrt{\eta}r)}{K_k(\sqrt{\eta}r)} + \sigma_1(k)}{\frac{\partial_r I_k(\sqrt{\eta}r)}{I_k(\sqrt{\eta}r)} + \sigma_1(k)}\Big|_{r=R+L} \cdot \frac{\frac{\partial_r I_k(\sqrt{\eta}r)}{I_k(\sqrt{\eta}r)} - \sigma_2(k)}{\frac{\partial_r K_k(\sqrt{\eta}r)}{K_k(\sqrt{\eta}r)} - \sigma_2(k)}\Big|_{r=R} \cdot \rho_{cla}. \tag{19}$$

We can see from (19) that the optimal choice for which ρ_{opt} vanishes is $\sigma_1(k) = -\frac{\partial_r K_k(\sqrt{\eta}r)}{K_k(\sqrt{\eta}r)}|_{r=R+L}$ and $\sigma_2(k) = \frac{\partial_r I_k(\sqrt{\eta}r)}{I_k(\sqrt{\eta}r)}|_{r=R}$, again the symbol of the non-local DtN operator. Optimized Schwarz methods use local approximations of the form

$$\sigma_i(k) = p_i + q_i k^2, \quad i = 1,2, \tag{20}$$

and determine p_i, q_i such that the convergence factor $\rho(k,L,R,\eta,p_1,p_2,q_1,q_2)$ is small. These transmission conditions are then easy to use and inexpensive. Simple approximations are obtained by Taylor expansion of the approximation $\sqrt{\eta+k^2/R_i^2}$ of the optimal symbol: T0 (Taylor of order zero) is given by $p_1=p_2=\sqrt{\eta}$, $q_1=q_2=0$, and leads with the estimate $k_{max}=\frac{\pi R}{h}$, where h is the mesh size, to the asymptotic convergence factor bounds $1-4\sqrt{2}\eta^{\frac{1}{4}}\sqrt{h}+O(h)$ with overlap $L=h$, and $1-4\sqrt{\eta}\pi^{-1}h+O(h^2)$ without overlap (still convergent!). T2 (Taylor of order two) is obtained with $p_i=\sqrt{\eta}$, $q_i=\frac{1}{2\sqrt{\eta}R_i}$, $i=1,2$, and leads to the bounds $1-8\eta^{\frac{1}{4}}\sqrt{h}+O(h)$ with overlap $L=h$, and $1-8\sqrt{\eta}\pi^{-1}h+O(h^2)$ without overlap. It is interesting to note that the curvature $1/R$ does not play a role in the asymptotic convergence factor estimates!

Optimized transmission conditions are based on minimizing the maximum of the convergence factor: let $C_{\text{OO0}}=\{p_1=p_2>0,q_1=q_2=0\}$, $C_{\text{OO2}}=\{p_1=p_2>0,q_1=q_2>0\}$ and $C_{\text{2-sided}}=\{p_1>0,p_2>0,q_1=q_2=0\}$. By solving the min-max problems

$$\min_{p_1,p_2,q_1,q_2\in C_I}\Big(\max_{k_{min}\le k\le k_{max}}|\rho(k,L,R,\eta,p_1,p_2,q_1,q_2)|\Big), \tag{21}$$

where the index $I\in\{\text{OO0},\text{OO2},\text{2-sided}\}$, we can determine the optimized choice of the parameters in each case. The corresponding optimized transmission conditions are then called OO0 (optimized of order 0), OO2 (optimized of order 2) and 2-sided (two-sided optimized) Robin transmission condition. Using asymptotic analysis, see [3] for details, we obtain for example for OO0 ($q_1=q_2=0$) $p_1=p_2=2^{-1}G_{min}^{\frac{2}{3}}h^{-\frac{1}{3}}$ and $\max_k|\rho_{OO0}|=1-4G_{min}^{\frac{1}{3}}h^{\frac{1}{3}}+O(h^{\frac{2}{3}})$ with overlap $L=h$, and $p_1=p_2=2^{-\frac{1}{2}}G_{min}^{\frac{1}{2}}\pi^{\frac{1}{2}}h^{-\frac{1}{2}}$ and $\max_k|\rho_{OO0}|=1-2^{\frac{3}{2}}G_{min}^{\frac{1}{2}}\pi^{-\frac{1}{2}}h^{\frac{1}{2}}+O(h)$ without overlap. Note that now also the convergence factor depends on the curvature $1/R$ through G_{min}. However, $\lim_{k_{min}\to 0}G_{min}=2\sqrt{\eta}$, independent of R.

4 Comparison of the two families of transmission conditions

We compare now the transmission conditions derived by micro-local analysis to the ones obtained based on optimization. We notice that MATC1 and T0 are identical; MATC5 looks like T2, but without the curvature dependence. In fact, MATC5 is exactly the Taylor condition of order 2 for a straight interface, see [4]. Next, we plot in Fig. 2 all the convergence factors of the Schwarz algorithm (17) with the various transmission conditions for a circular decomposition. We observe that MATC2-4 perform similarly to T2. Since MATC2-4 are of order ≤ 1, we conclude that involving the curvature does improve the performance. It also seems that MATC5 performs quite well. However, this is not always the case: we show a comparison between the three second order transmission conditions in Fig. 3. We can see that MATC5 is much more sensitive to R ($1/R$ is the curvature) than the other two, both in the case

Table 1 Number of iterations required by the Schwarz algorithm with different transmission conditions with overlap $L = h$ and without overlap (in parentheses)

h	Cla	MATC1(T0)	MATC2	MATC3	MATC4	MATC5	T2	OO0	OO2	2-sided
1/50	332	26(310)	20(177)	20(173)	22(208)	17(370)	18(1081)	16(52)	14(48)	41(41)
1/100	684	36(597)	29(354)	27(331)	32(410)	16(644)	23(1832)	21(75)	13(57)	35(51)
1/200	1279	51(1163)	40(662)	39(646)	42(784)	17(1033)	29(3048)	26(101)	14(62)	27(61)
1/400	2919	71(2236)	53(1296)	53(1236)	59(1519)	22(1536)	39(4294)	32(151)	14(70)	23(71)

with and without overlap: the optimized transmission condition performs always better than the other two; the MATC5 gets its best performance around $R = 0.5$ (this is exactly the case of Fig. 2), it performs as T2 at $R = 1$, since then the approximation is identical, and with increasing R it performs worse and worse. We finally note that MATC6 does not perform well: in the middle of Fig. 2, we see that near $k = 1.5$ the convergence factor blows up. Hence MATC6 is not a good choice as transmission condition.

5 Numerical experiments

We perform numerical experiments for a model problem in polar coordinates,

$$\begin{aligned} \partial_{rr}u + \tfrac{1}{r}\partial_r u + \tfrac{1}{r^2}\partial_{\theta\theta}u - \eta u &= f(r,\theta) \quad \text{in } \Omega, \\ u &= 0 \qquad \text{on } \partial\Omega, \end{aligned} \tag{22}$$

where $\Omega = (0,1)\times(0,2\pi)$ is decomposed into $\Omega = \Omega_1 \cup \Omega_2$, with $\Omega_1 = (0, R+L)\times(0,2\pi)$ and $\Omega_2 = (R,1)\times(0,2\pi)$, and $L \geq 0$ is the overlap. We use a finite difference scheme on a uniform grid with mesh size h to simulate directly the error equations, $f = 0$, for $R = 0.5$ and $\eta = 2$, and a random initial guess is chosen so that all the frequency components are present in the initial error. The number of iterations required by the parallel Schwarz method (17) are shown in Table 1. We clearly see

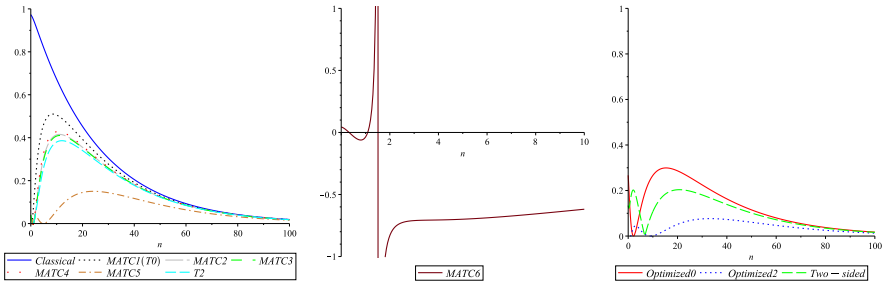


Fig. 2 Convergence factors of MATC1-5 and the Taylor conditions (left), MATC6 (middle), and with optimized transmission conditions (right), for $\eta = 2$, overlap $L = 0.01$ and $R = 0.5$

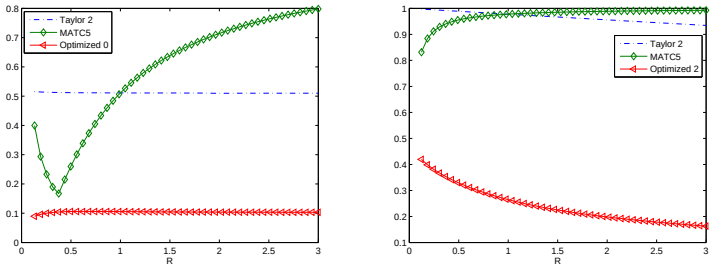


Fig. 3 The maxima of the convergence factors as functions of R with overlap (left) and without (right)

that the transmission conditions based on optimization get better performance in this experiment.

6 Conclusion

We presented two different approaches to take the curvature of interfaces into account in the transmission conditions of optimized Schwarz methods: micro-local analysis, and analysis using a circular model problem. In both cases, we obtained curvature dependent transmission conditions. A preliminary comparison shows that the transmission conditions based on optimization perform better on the model problem, and that it could be important to take the curvature into account in transmission conditions. In our opinion it is however essential to do a more thorough theoretical and numerical study on more general geometry, where micro-local analysis is still applicable, before we can definitely draw conclusions.

References

1. Abramowitz, M., Stegun, I.A.: Handbook of mathematical functions with formulas, graphs, and mathematical tables. Dover pulications, New York (1972)
2. Antoine, X., Barucq, H., Bendali, A.: Bayliss-Turkel-like radiation conditions on surfaces of arbitrary shape. J. Math. Anal. Appl. **229**(1), 184–211 (1999)
3. Barucq, H., Gander, M., Xu, Y.: Optimized Schwarz methods for a reaction-diffusion model with circular domain decomposition. in preparation. (2013)
4. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**(2), 699–731 (2006)
5. Lui, S.H.: A Lions non-overlapping domain decomposition method for domains with an arbitrary interface. IMA J. Numer. Anal. **29**(2), 332–349 (2009)
6. Lui, S.H.: Convergence estimates for an higher order optimized Schwarz method for domains with an arbitrary interface. J. Comput. Appl. Math. **235**(1), 301–314 (2010)

Conservative inexact solvers for porous media flow

Eirik Keilegavlen[1] and Jan M. Nordbotten[1]

1 Introduction

Simulation models of flow and transport in geological porous media are characterized by a high degree of uncertainty due to both discretization errors and incomplete measurements of physical parameters. In the context of linear solvers this seemingly mandates the use of inexact strategies, where a solution is sought with an accuracy similar to that of the overall computational model. Since the solution of linear systems often consumes a substantial part of the total simulation time, inexact solvers can yield considerable computational savings. However the derivation of the continuous model is based on conservation of mass, and this property must be preserved in the discrete system for the results to be physically meaningful. The discretization schemes commonly applied are conservative by construction, but unless the linear solver is designed specifically to produce solutions that, even if inexact, conserve mass the inexact solution may not yield a stable overall simulation strategy. For this reason linear systems are commonly solved to an accuracy that is much higher than mandated by known discretization errors and parameter uncertainties.

The key to producing physically meaningful inexact solutions is to design the linear solver by the same principles as the discretization scheme. Herein we will explore these ideas in the context of two-phase flow in a horizontal porous media. The phases denoted water (w) and oil (o) are immiscible and incompressible with a velocity given by Darcy's law

$$\mathbf{u}_\alpha = -\lambda_\alpha \mathbf{K} \nabla p, \qquad \alpha = w, o. \tag{1}$$

Here the phase mobilities λ_w and λ_o, represent fluid viscosity and rock-fluid interaction. Furthermore $\mathbf{K}$ is the permeability and p is the fluid pressure. Of particular importance to this paper are the properties of the permeability, which commonly possesses sharp contrasts of several orders of magnitude and spatial correlation structures on a continuum of length scales. Conservation of mass for each phase is expressed as

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot \mathbf{u}_\alpha = q_\alpha, \qquad \alpha = w, o, \tag{2}$$

where ϕ represents porosity, S_α is the volume fraction of phase α and q_α is the source term. The saturations are assumed to fill the pore volume, that is $S_w + S_o = 1$. Thus when (2) for the two phases are added to get an equation for conservation of

[1]Department of Mathematics, University of Bergen, e-mail: {Eirik.Keilegavlen}{Jan.Nordbotten}@math.uib.no

total mass, the saturations are eliminated. This gives a linear elliptic equation for the pressure, which can be written

$$\nabla \cdot \mathbf{u}_T = -\nabla \cdot (\lambda_T \mathbf{K} \nabla p) = q_T. \tag{3}$$

Here $\mathbf{u}_T = \mathbf{u_w} + \mathbf{u_o}$ is the total velocity, $\lambda_T = \lambda_w + \lambda_o$ is the total mobility and $q_T = q_w + q_o$ is the total source term.

2 Discretization

In the rest of the paper we describe the construction of an inexact linear solver for (3) which preserves the conservation property of $\mathbf{u}_T$. The solver is formulated in terms of a novel multi-level control volume method which is briefly described next. More details can be found in [6].

2.1 A hierarchy of control volume discretizations

In applications conservation of mass is considered an essential property that should be preserved during discretization. To that end a cell centered control volume method is applied for the spatial discretization. A discrete Darcy's law is constructed as in [1]

$$\mathbf{u}_{h,\alpha} = -\lambda_\alpha^U T_h p_h, \tag{4}$$

where $\mathbf{u}_{h,\alpha}$ is the discrete phase velocities for phase α, T_h is a matrix of transmissibilities and p_h is a cell centered approximation of the pressure. The mobilities, λ_α^U, are discretized by phase-wise upstream weighting. A discrete equation for the pressure is found by

$$D_h((\lambda_w^U + \lambda_o^U) T_h p_h) = A_h p_h = q_h, \tag{5}$$

where D_h is the discrete divergence, A_h is the system matrix and q_h represents discrete sources. We note that (5) can be considered a Petrov-Galerkin discretization of (3), with piece-wise constants on the cells as test functions and shape functions defined by the specific control volume method. When (5) has been solved for p_h, (2) for the water phase is discretized by an explicit method with upstream weighting of the mobilities.

The sharp contrasts and long correlation structures of the permeability is reflected in the discretization matrix A_h, thus solving (5) is time consuming. Discretization errors and uncertainties in the permeability make the linear system a prime candidate for an inexact linear solver. However, (5) was derived by requiring conservation of mass, and unless this is reflected in the inexact solution, conservation errors will

in worst case grow exponentially in the time propagation of (2). The linear solver should therefore be constructed to produce a discrete flux field that, even if inexact, satisfies (5). Furthermore an efficient solution strategy for (5) should invoke coarse solvers to account for the global dependencies of the equation.

An inexact two-level method which retains the conservation property can be realized within the framework of the multiscale finite volume (MSFV) method [3], see also [7]. The domain is partitioned into a coarse grid and a coarse shape function ψ_H is constructed for each coarse cell to account for fine-scale variabilities in the permeability. Coarse test functions ϕ_H are defined as piece-wise constants on the coarse cells. A coarse linear system is then defined as

$$(\Phi_H^T A_h \Psi_H) p_H = A_H p_H = \Phi_H^T q_h. \tag{6}$$

Here Φ_H and Ψ_H are column matrices of test and shape functions, respectively, and A_H is the coarse discretization. It is important to note the similarity between (5) and (6), in that both are obtained by applying Petrov-Galerkin techniques. In this way the coarse linear system retains the conservation property of the fine-scale discretization. Specifically it will produce conservative coarse fluxes in the sense that the fluxes into a coarse cell match the sources within the cell. When projected to the fine scale the inexact fluxes will not be conservative. This is remedied by a post-processing step where local fine-scale problems are solved within each coarse cell [3]. The boundary conditions are the projection of the conservative coarse fluxes to the fine scale.

2.2 Multi-level flux post-processing

The two-level method outlined above amounts to an inexact linear solver that can also be applied as a preconditioner within an iterative solver. However it is natural to seek multi-level methods to realize efficient residual smoothing strategies. Also when multiple grid levels are available, adaptive upscaling can be applied during the simulation. Finally, the MSFV method is known to be unstable in cases where the coarse grid does not follow anisotropy patterns in the permeability [5]. This can be remedied by an unstructured coarsening strategy that is currently under development but for this approach to be robust multiple coarsening steps with mild upscaling ratios should be applied.

Since (6) has the same properties as (5) in terms of sparsity pattern and conservation property, a further coarsening of the system can easily be constructed by recursion. However, for the multi-level method to be applicable as a conservative inexact linear solver, multi-level post-processing is needed, and specifically local Neumann problems must be solved. For the coarser levels the discretization of Neumann boundary conditions is not available, and this has in practice limited control

volume linear solvers to two grid levels. In the following we will outline how the multi-level post-processing can be realized, a thorough explanation is given in [6].

As for the two-level method, the post-processing is performed by solving local problems that are confined to single cells on the coarser level. When conservative fluxes on coarse faces are known these can be mapped to any finer level via the shape functions, specifically they can be mapped one level down to form boundary conditions for the local problems. In this way the flux discretization on coarse boundaries is replaced by known fluxes. However there will be faces interior to the coarse cell with exterior cells in their flux discretization, in conflict with the goal of a local post-processing. The exterior cells are eliminated by considering groups of cells that are centered around vertexes on the boundary of the coarse cell and have common support for their basis functions, as illustrated in Fig. 1. The exterior cells can be replaced by the known fluxes over the boundary by formulating and solving a local linear system. When the number of exterior cells and the number of known fluxes are equal, the elimination is straightforward. If there are more exterior cells than there are boundary conditions (respectively 3 and 2 in Fig. 1), additional equations can be obtained by splitting the boundary fluxes into sub-fluxes on a finer grid level and computing higher order moments of the fluxes based on these. Note that on the finest level the elimination is straightforward since a boundary discretization is available there; thus the splitting into sub-fluxes is available when needed. A linear system is then solved around all vertexes on the boundary, and the results are used to formulate a local system within the coarse cell that is solved to get conservative fluxes.

This methodology provides conservative fluxes for all faces on all grid levels even if the accompanying pressure is inexact. We make two comments on the approach: firstly the only pair of pressure and fluxes which satisfies both the discrete flux law (4) and the conservation equation (5) is the exact solution. The post-processed fluxes possess the conservation property, but they cannot be computed from the inexact pressures via (4). The post-processed fluxes can be thought of as being exact for a modified permeability field, in accordance with an uncertainty in this parameter. Secondly the post-processing is not applicable unless the inexact solution preserves the conservation property of the continuous problem. This not only

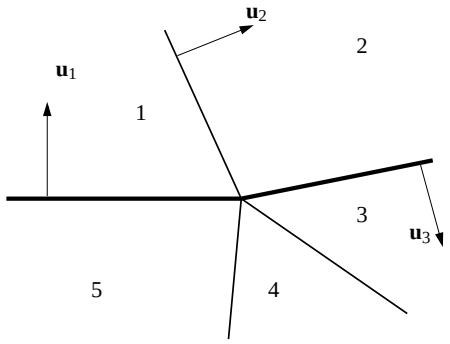


Fig. 1 Parts of cells with common support for their basis functions centered around a vertex at the boundary of a coarse cell. Fluxes (arrows) and cells close to the boundary of a coarse cell (bold). Cells 3-5 are outside the coarse cell and must be eliminated from the flux expression for $\mathbf{u}_2$ using $\mathbf{u}_1$ and $\mathbf{u}_3$ (which are known) and their sub-fluxes.

requires the construction of coarse problems as described above, but also a careful treatment of the right hand side of the linear system. To be specific, the right hand side should be coarsened according to the Schur complement formulation of the multi-level method [8]. The multi-level method with this special coarsening can be applied as a correction to the residual of any inexact solution. The corrected solution will in general still be inexact, but it will possess the structure necessary to apply the post-processing.

2.3 Error control

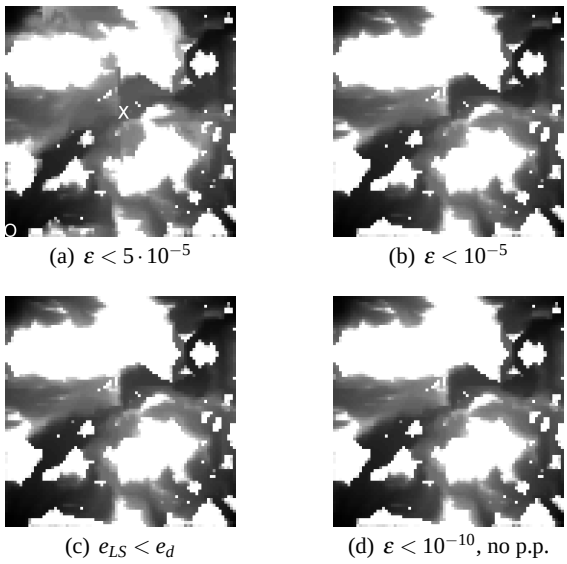
With the post-processing outlined above, we can obtain solutions that are inexact but still honor the conservation property. There are two natural criteria for controlling the linear solver. The simplest option is to terminate the iterations when a desired reduction of the relative residual is achieved and then apply post-processing to obtain a mass conserving flux field. However, even though the post-processing produces a velocity field without conservation errors a reduction of the relative residual gives little control of the accuracy of the fluxes. A more nuanced notion of error can be derived from [4], where we find the expression

$$\|\mathbf{K}^{-1/2}(\mathbf{u}-\mathbf{u}_h^*)\| \leq \inf_{s\in H^1} \|\mathbf{K}^{-1/2}(\mathbf{u}_h^*-\mathbf{K}\nabla s)\| + \sup_{\beta\in H^1,\|\beta=1\|} (\nabla\cdot(\mathbf{u}-\mathbf{u}_h^*),\beta), \quad (7)$$

where $\mathbf{u}$ is the true flux and $\mathbf{u}_h^*$ is the post-processed flux field. The last term evaluates to zero since the post-processed and exact fluxes have the same divergence. The triangular inequality applied on the first term gives

$$\|\mathbf{K}^{-1/2}(\mathbf{u}-\mathbf{u}_h^*)\| \leq \|\mathbf{K}^{-1/2}(\mathbf{u}_h^*-\mathbf{K}\nabla p_h^*)\| + \inf_{s\in H^1} \|\mathbf{K}^{1/2}\nabla(p_h^*-s)\|, \quad (8)$$

with p_h^* representing the inexact pressure. The first term on the right hand side of (8) is immediately computable, and can be interpreted as the error stemming from the linear solver. We denote this term e_{LS}. The second term is identified as the discretization error, denoted e_d. To give reasonable estimates for the gradient of p_h^* in heterogeneous media, we compute this from face pressures that are reconstructed from the fine-scale discretization. The estimate (8) can employed to control the linear solution process by terminating the iterations when the error from the linear solver is smaller than the discretization error, at which point it can be argued that it makes little sense to improve the inexact solution.



(a) $\varepsilon < 5 \cdot 10^{-5}$ (b) $\varepsilon < 10^{-5}$

(c) $e_{LS} < e_d$ (d) $\varepsilon < 10^{-10}$, no p.p.

Fig. 2 Saturation profiles obtained with different stopping criteria for the linear solvers. Water (light) is injected into a domain initially filled with oil (dark). Injection (O) and production (X) wells are marked in (a). Periodic boundary conditions are applied.

3 Numerical results

In this section we illustrate the utility of the conservative framework by coupling an inexact multi-level linear solver for the pressure equation to a non-linear transport problem. The computational grid is Cartesian, with 3^4 cells in each direction. The permeability is taken from the bottom layer of the 10th SPE comparative solution project (SPE10) [2], which is characterized by long and highly permeable channels and sharp contrasts of 6 orders of magnitude. The medium is initially filled with oil. Water is injected in the lower left corner of the grid, and a production well is placed in the middle of the domain.

The phase velocities in (4) are discretized on the fine-scale grid by a two-point flux approximation. Periodic boundary conditions are assigned for simplicity. Three levels of coarsening are applied, each with a ratio of 3 in each direction, and a direct solver is invoked on the coarsest grid. Thus the coarse operator constitutes a four-level multi-grid method. Updates of the saturation feed back to the pressure equation via the mobilities, which are set to $\lambda_w = S_w^3$ and $\lambda_o = 10S_o^2$, and thus the velocity field must be updated regularly. The pressure time step is fixed at a tenth of the total simulation time, while the time step for the saturation equation is decided by the CFL criterion.

To solve the pressure equation, GMRES iteration preconditioned by the multi-level method is applied. Four criteria for terminating the iterative solver are considered: Two consider the reduction of the relative residual, ε, and terminate the

iterations when $\varepsilon < 5 \cdot 10^{-5}$ and $\varepsilon < 10^{-5}$, respectively. The third criterion requires that $e_{LS} < e_d$, which in this case corresponds to a value of ε of $10^{-6} - 10^{-8}$. All these estimates apply post-processing to ensure the approximated flux field is conservative. Finally, we consider a solver with the same preconditioner, but where post-processing is not applied after the iterations. In this case the fluxes must be brought sufficiently close to being conservative by iterating on the solution. Note that this is the strategy applied by a traditional linear solver. For the present setup, a value of $\varepsilon < 10^{-10}$ is needed to avoid severe stability issues due to conservation errors.

Table 1 Total number of GMRES iterations needed to achieve desired tolerance level.

$\varepsilon < 5 \cdot 10^{-5}$	$\varepsilon < 10^{-5}$	$e_{LS} < e_d$	$\varepsilon < 10^{-10}$, no p.p.
190	200	212	293

Snapshots of the saturation distributions with the respective control parameters are shown in Fig. 2. All simulations predict the same large-scale pattern, and it is only the loosest tolerance for the pressure solver that yields notable differences in the saturation profile. The computational gains from applying post processing can be seen from the number of iterations shown in Tab. 1. We observe that there is considerable room for computational savings without sacrificing significant accuracy of the transport solution. We reiterate that this is due to the post-processing, which facilitates inexact yet conservative flux fields. Some caution is needed when deciding the stopping criterion for the linear solver as the accuracy necessary to get reasonable transport solution is highly dependent on the simulation setup. Note that if the post-processing is not applied the accuracy to produce a flux field that makes the transport solver behaves stable increases significantly. The tolerance necessary will be different for other simulations, and in practice the only options to obtain stable simulations are to iterate until the exact solution is found, or to apply an inexact solver and somehow tackle conservation errors in the transport solver. We also remark that the performance of all preconditioners suffers from the Cartesian coarse grids that leads to strong heterogeneities within the coarse cells. This will be amended by an unstructured coarsening procedure currently under development.

4 Concluding remarks

In this paper we have considered the application of an inexact linear solver for porous media flow with the special property that it provide a set of fluxes that exactly satisfy a conservation law, even if the associated pressure that drives the flux was approximated. The solver was formulated as a multi-level control volume discretization, and we considered the coupling of the solver with a non-linear transport problem. Since the approximated flux field possessed the conservation prop-

erty, considerable computational savings were possible without sacrificing stability or significant accuracy in the transport simulation.

For simulation of realistic applications there will always be a trade-off between accuracy and computational effort, and this balance is particularly well articulated when control parameters for linear solvers are decided. We have shown in this paper that the linear solver should not be considered a stand-alone part of the overall simulation tool. Instead it should be in accordance with the same principles as guided the choice of the disrcetization scheme. The resulting solver will provide solutions that even if approximated are physically meaningful, enhancing the robustness of the simulator.

References

1. Aziz, K., Settari, A.: Petroleum Reservoir Simulation. Chapman & Hall (1979). ISBN: 0-85334-787-5
2. Christie, M., Blunt, M.: Tenth SPE comparative solution project: A comparison of upscaling techniques. SPEREE **4**, 308–317 (2001)
3. Jenny, P., Lee, S.H., Tchelepi, H.A.: Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. J. Comput. Phys. **187**(1), 47–67 (2003)
4. Jiránek, P., Strakoš, Z., Vohralík, M.: A posteriori error estimates including algebraic error and stopping criteria for iterative solvers. SIAM J Sci. Comput. **32**(3), 1567–1590 (2010)
5. Lunati, I., Jenny, P.: Treating highly anisotropic subsurface flow with the multiscale finite-volume method. Multiscale Model. Simul. **6**(1), 308–318 (2007)
6. Keilegavlen, E., Nordbotten, M.: Inexact linear solvers for control volume discretizations. Submitted to Comput. Geosci.
7. Nordbotten, J., Keilegavlen, E., Sandvin, A.: Finite volume method - Powerful means of engineering design, chap. Mass Conservative Domain Decomposition for Porous Media Flow. In Tech (2012). 978-953-51-0445-2
8. Nordbotten, J.M., Bjørstad, P.E.: On the relationship between the multiscale finite-volume method and domain decomposition preconditioners. Comput. Geosci. **12**(3), 367–376 (2008)

Robust isogeometric Schwarz preconditioners for composite elastic materials

L. Beirão da Veiga[1], D. Cho[2], L. F. Pavarino[1], and S. Scacchi[1]

1 Introduction

In this paper, we study Overlapping Schwarz preconditioners for the system of linear elasticity for composite materials discretized with Isogeometric Analysis (IGA). This is an innovative numerical methodology, introduced by Hughes et al. [10, 6], where the geometry description of the PDE domain is adopted from a Computer Aided Design (CAD) parametrization usually based on Non-Uniform Rational B-Splines (NURBS). In IGA, these NURBS basis functions representing the CAD geometry are also used as the PDEs discrete basis, following an isoparametric paradigm. Since its introduction, IGA techniques have been studied and applied in diverse fields, see e.g. [6].

In our previous Domain Decomposition (DD) works for IGA scalar elliptic problems, we studied Overlapping Additive Schwarz (OAS) methods [2] and Balancing Domain Decomposition by Constraints (BDDC) methods [3], providing optimal and quasi-optimal convergence rate bounds for isogeometric DD methods, together with the required theoretical foundation, technical tools and numerical validation. Other DD IGA works have explored numerically dual primal Finite Element Tearing and Interconnecting (FETI-DP) methods for 2D elliptic problems [11] and have studied multigrid methods for the 2D and 3D Laplacian [9] and Schwarz methods in the case of two subdomains with non-matching grids [5].

Here we study Isogeometric OAS preconditioners for the system of linear elasticity for compressible composite materials. An extension to mixed methods for almost incompressible elastic materials can be found in [4].

We consider the linear elastic deformation of a body Ω in $\mathbb{R}^d$, $d = 2,3$, with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$. The body is clamped on Γ_D and it is subjected to a given traction $g : \Gamma_N \to \mathbb{R}^d$ on Γ_N, as well as to a body force density $f : \Omega \to \mathbb{R}^d$. The displacement field $u : \Omega \to \mathbb{R}^d$ satisfies the system

$$\begin{cases} \operatorname{div} \mathbb{C}\varepsilon(u) + f = 0 & \text{in } \Omega \\ u = 0 \text{ on } \Gamma_D \quad \text{and} \quad \mathbb{C}\varepsilon(u) \cdot n = g \text{ on } \Gamma_N \end{cases} \tag{1}$$

[1]Department of Mathematics, University of Milano, via Saldini 50, 20133 Milano, Italy, e-mail: {lourenco.beirao}{luca.pavarino}{simone.scacchi}@unimi.it .[2] Department of Mathematics, Dongguk University, Pil-dong 3-ga, Jung-gu, Seoul, 100-715, South Korea, e-mail: durkbin@dongguk.edu

Here, ε is the symmetric gradient operator, n is the unit outward normal at each point of the boundary, $\mathbb{C}\tau = 2\mu\tau + \lambda \mathrm{tr}(\tau) I$ for all second order tensors τ, where $\mathrm{tr}(\tau)$ is the trace of τ, $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \mu = \frac{E}{2(1+\nu)}$ are the Lamé constants, E is the Young modulus and ν the Poisson's ratio. Given loadings $f \in [L^2(\Omega)]^d$ and $g \in [L^2(\Gamma_N)]^d$, define

$$< \psi, v >= (f,v)_\Omega + (g,v)_{\Gamma_N} \qquad \forall v \in [H^1(\Omega)]^d, \tag{2}$$

where $(\cdot,\cdot)_\Omega$, $(\cdot,\cdot)_{\Gamma_N}$ indicate as usual the L^2 scalar product respectively on Ω and Γ_N. The variational formulation of problem (1) then reads:

$$\begin{cases} \text{Find } u \in [H^1_{\Gamma_D}(\Omega)]^d \text{ such that:} \\ a(u,v) =< \psi, v > \qquad \forall v \in [H^1_{\Gamma_D}(\Omega)]^d, \end{cases} \tag{3}$$

where $[H^1_{\Gamma_D}(\Omega)]^d = \{v \in [H^1(\Omega)]^d \,|\, v|_{\Gamma_D} = 0\}$ and

$$a(w,v) = \int_\Omega 2\mu\, \varepsilon(w) : \varepsilon(v)\, dx + (\lambda\, \mathrm{div} w, \mathrm{div} v)_\Omega \qquad \forall w, v \in [H^1_{\Gamma_D}(\Omega)]^d. \tag{4}$$

2 Isogeometric discretization of linear elasticity

We discretize the elasticity system (3) with IGA based on B-splines and NURBS basis functions, see e.g. [6]. Considering for simplicity the two-dimensional case, the bivariate B-spline discrete space is defined as

$$\widehat{\mathscr{S}_h} = \mathrm{span}\{B^{p,q}_{i,j}(\xi,\eta),\ i = 1,\dots,n,\ j = 1,\dots,m\}, \tag{5}$$

where the bivariate B-spline basis functions $B^{p,q}_{i,j}(\xi,\eta) = N^p_i(\xi)\, M^q_j(\eta)$ are defined by tensor product of one-dimensional B-splines functions $N^p_i(\xi)$ and $M^q_j(\eta)$ of degree p and q, respectively. Analogously, the NURBS space is the span of NURBS basis functions defined in 1D as

$$R^p_i(\xi) = \frac{N^p_i(\xi)\omega_i}{\sum_{\hat{\imath}=1}^n N^p_{\hat{\imath}}(\xi)\omega_{\hat{\imath}}} = \frac{N^p_i(\xi)\omega_i}{w(\xi)}, \tag{6}$$

(with weight function $w(\xi) = \sum_{\hat{\imath}=1}^n N^p_{\hat{\imath}}(\xi)\omega_{\hat{\imath}} \in \widehat{\mathscr{S}_h}$), and in 2D by tensor product

$$R^{p,q}_{i,j}(\xi,\eta) = \frac{B^{p,q}_{i,j}(\xi,\eta)\omega_{i,j}}{\sum_{\hat{\imath}=1}^n \sum_{\hat{\jmath}=1}^m B^{p,q}_{\hat{\imath},\hat{\jmath}}(\xi,\eta)\omega_{\hat{\imath},\hat{\jmath}}} = \frac{B^{p,q}_{i,j}(\xi,\eta)\omega_{i,j}}{w(\xi,\eta)}, \tag{7}$$

where $w(\xi,\eta)$ is the weight function and $\omega_{i,j} = (C^\omega_{i,j})_3$ the weights associated with a $n \times m$ net of control points $C_{i,j}$. The discrete space of NURBS scalar fields on the domain Ω is defined, component by component as the span of the *push-forward* of the NURBS basis functions (7)

$$\mathscr{N}_h := \text{span}\{R_{i,j}^{p,q} \circ F^{-1}, \text{ with } i = 1,\dots,n; j = 1,\dots,m\}, \tag{8}$$

with $F : \widehat{\Omega} \to \Omega$ the geometrical map between parameter and physical spaces

$$F(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta) C_{i,j}. \tag{9}$$

Taking into account the boundary conditions, if for simplicity we consider the case $\Gamma_D = \partial\Omega$, we define the spline space in parameter space as

$$\widehat{V}_h = [\widehat{\mathscr{S}_h} \cap H_0^1(\widehat{\Omega})]^d = [\text{span}\{B_{i,j}^{p,q}(\xi,\eta),\ i = 2,\dots,n-1,\ j = 2,\dots,m-1\}]^d.$$

and the NURBS space in physical space as

$$V_h = [\mathscr{N}_h \cap H_0^1(\Omega)]^d = [\text{span}\{R_{i,j}^{p,q} \circ F^{-1}, \text{ with } i = 2,\dots,n-1;\ j = 2,\dots,m-1\}]^d. \tag{10}$$

The IGA formulation of problem (3) then reads:

$$\begin{cases} \text{Find } u_h \in V_h \text{ such that:} \\ a(u_h, v_h) =< \psi, v_h > \qquad \forall v_h \in V_h. \end{cases} \tag{11}$$

3 Isogeometric Overlapping Schwarz preconditioners

We refer to the monographs [12, 13] for a general introduction to Overlapping Schwarz methods. We describe first the subdomain and subspace decompositions in 1D and then extend them by tensor products to 2D and 3D. The decomposition is first built for the underlying space of spline functions in parameter space, and then easily extended to the NURBS space in the physical domain.

1D B-spline decomposition. From the full set of knots $\{\xi_1 = 0, ..., \xi_{n+p+1} = 1\}$, we select a subset $\{\xi_{i_k}, k = 1,\dots,N+1\}$ of (non repeated) interface knots with $\xi_{i_1} = 0, \xi_{i_{N+1}} = 1$. This subset of interface knots defines a decomposition of the closure of the reference interval

$$\overline{(\widehat{I})} = [0,1] = \overline{\Big(\bigcup_{k=1,..,N} \widehat{I}_k\Big)}, \text{ with } \widehat{I}_k = (\xi_{i_k}, \xi_{i_{k+1}}),$$

that we assume to have a similar characteristic diameter $H \approx H_k = \text{diam}(\widehat{I}_k)$. The interface knots are thus given by ξ_{i_k} for $k = 2,..,N$. For each of the interface knots ξ_{i_k} we choose an index $2 \le s_k \le n-1$ (strictly increasing in k) that satisfies $s_k < i_k < s_k + p + 1$, so that the support of the basis function $N_{s_k}^p$ intersects both $\widehat{I}_{k-1}$ and $\widehat{I}_k$. Note that at least one such s_k exists; if it is not unique, any choice can be made.

We then define an overlapping decomposition of $\widehat{I}$ in the following way. Let $r \in \mathbb{N}$ be an integer (called the overlap index) counting the basis functions shared by

adjacent subdomains, defined as

$$\widehat{V}_k = [\text{span}\{N_j^p(\xi),\ s_k - r \le j \le s_{k+1} + r\}]^d \qquad k = 1,2,..,N, \tag{12}$$

with the exception that $2 \le j \le s_2 + r$ for the space $\widehat{V}_1$ and $s_N - r \le j \le n-1$ for the space $\widehat{V}_N$. These subspaces form an overlapping decomposition of the spline space $\widehat{V}_h$. For $r = 0$ we have the minimal overlap consisting of just one common basis function between subspaces, while more generally $2r+1$ represents the number of basis functions in common (in the univariate case) among "adjacent" local subspaces. We now define the extended subdomains $\widehat{I}_k'$ by

$$\widehat{I}_k' = \bigcup_{N_j^p \in \widehat{V}_k} \text{supp}(N_j^p) = (\xi_{s_k - r}, \xi_{s_{k+1}+r+p+1}), \tag{13}$$

with the analogous exception for $\widehat{I}_1'$, $\widehat{I}_N'$,

We consider two choices for the *coarse space* $\widehat{V}_0$.

a) A nested coarse space defined by introducing a (open) coarse knot vector $\xi_0 = \{\xi_1^0 = 0, ..., \xi_{N_c+p+1}^0 = 1\}$ corresponding to a coarse mesh determined by the subdomains $\widehat{I}_k$, i.e.

$$\xi_0 = \{\xi_1, \xi_2, \ldots, \xi_p, \xi_{i_1}, \xi_{i_2}, \xi_{i_3}, \ldots, \xi_{i_N}, \xi_{i_{N+1}}, \xi_{i_{N+1}+1}, \xi_{i_{N+1}+2} \ldots, \xi_{i_{N+1}+p}\},$$

such that the distance between adjacent distinct knots is of order H, $\xi_1 = \cdots = \xi_p = \xi_{i_1} = 0$ and $\xi_{i_{N+1}} = \xi_{i_{N+1}+1} = \cdots = \xi_{i_{N+1}+p} = 1$. A coarse spline space is then defined as

$$\widehat{V}_0 := [\widehat{\mathscr{S}_H}]^d = [\text{span}\{N_i^{0,p}(\xi),\ i = 2, ..., N_c - 1\}]^d,$$

with the same degree p of $\widehat{\mathscr{S}_h}$ and is thus a subspace of $[\widehat{\mathscr{S}_h}]^d$.

b) A non-nested coarse space, of smaller dimension than in case a), is defined as

$$\widehat{V}_0 := [\widehat{\mathscr{S}_H}]^d = [\text{span}\{N_i^{0,1}(\xi),\ i = 2, ..., N_c - 1\}]^d,$$

where now note that $p = 1$ and the coarse knot vector (and N_c) is changed accordingly

$$\xi_0 = \{\xi_1, \xi_{i_1}, \xi_{i_2}, \xi_{i_3}, \ldots, \xi_{i_N}, \xi_{i_{N+1}}, \xi_{i_{N+1}+1}\},$$

with $\xi_1 = \xi_{i_1} = 0$ and $\xi_{i_{N+1}} = \xi_{i_{N+1}+1} = 1$. The construction above gives the standard piecewise linear space on the coarse subdivision.

2D, 3D B-spline decomposition. By tensor product (here in 2D for simplicity), we define subdomains, overlapping subdomains and extended supports by

$$\widehat{\Omega}_{kl} = \widehat{I}_k \times \widehat{I}_l, \qquad \widehat{\Omega}_{kl}' = \widehat{I}_k' \times \widehat{I}_l', \qquad 1 \le k \le N,\ 1 \le l \le M,$$

(where $\widehat{I}_k = (\xi_{i_k}, \xi_{i_{k+1}}), \widehat{I}_l = (\eta_{j_l}, \eta_{j_{l+1}})$). Moreover, we take the indices $\{s_k\}_{k=2}^N$ associated to $\{\xi_{i_k}\}_{k=2}^N$ and the analogous indices $\{\bar{s}_l\}_{l=2}^M$ associated to $\{\eta_{j_l}\}_{l=2}^M$. The local and coarse subspaces are then defined by

$$\widehat{V}_{kl} = [\text{span}\{B_{i,j}^{p,q}(\xi,\eta),\ s_k - r \le i \le s_{k+1} + r,\ \bar{s}_l - r \le j \le \bar{s}_{l+1} + r\ \}]^d,$$
$$\widehat{V}_0 = [\text{span}\{\overset{\circ}{B}{}_{i,j}^{p,q}\ :\ \overset{\circ}{B}{}_{i,j}^{p,q}(\xi,\eta) := N_i^{0,p}(\xi) M_j^{0,q}(\eta),\ i = 1,...,N_c, j = 1,...,M_c\}]^d,$$

with the usual modification for boundary subdomains and where $\overset{\circ}{B}{}_{i,j}^{p,q}$ are the coarse basis functions.

2D, 3D NURBS decomposition. The subdomains in physical space are defined as the image of the subdomains in parameter space with respect to the mapping F

$$\Omega_{kl} = F(\widehat{\Omega}_{kl}), \quad \Omega'_{kl} = F(\widehat{\Omega}'_{kl}).$$

The local subspaces and the coarse space are, up to the usual modification for the boundary subdomains,

$$V_{kl} = [\text{span}\{R_{i,j}^{p,q} \circ F^{-1},\ s_k - r \le i \le s_{k+1} + r,\ \bar{s}_l - r \le j \le \bar{s}_{l+1} + r\ \}]^d,$$
$$V_0 = [\text{span}\{\overset{\circ}{R}{}_{i,j}^{p,q} \circ\ F^{-1} := (\overset{\circ}{B}{}_{i,j}^{p,q}/w) \circ\ F^{-1},\ i = 1,...,N_c, j = 1,...,M_c\}]^d,$$

where we recall that w is the weight function, see (7).

Overlapping Schwarz preconditioners. Given the local and coarse embedding operators $I_{kl} : V_{kl} \to V_h$, $k = 1,..,N$, $l = 1,..,M$ and $I_0 : V_0 \to V_h$, the discrete space V_h can be decomposed into coarse and local space as

$$V_h = I_0 V_0 + \sum_{k,l} I_{kl} V_{kl}.$$

Define the local projections $\widetilde{T}_{kl} : V_h \to V_{kl}$ by

$$a(\widetilde{T}_{kl} u, v) = a(u, I_{kl} v) \quad \forall v \in V_{kl},$$

and the coarse projection $\widetilde{T}_0 : V_h \to V_0$ by

$$a(\widetilde{T}_0 u, v) = a(u, I_0 v) \quad \forall v \in V_0.$$

Defining $T_{kl} = I_{kl}\widetilde{T}_{kl}$ and $T_0 = I_0\widetilde{T}_0$, our two-level Overlapping Additive Schwarz (OAS) operator is then

$$T_{OAS} := T_0 + \sum_{k=1}^{N}\sum_{l=1}^{M} T_{kl}. \tag{14}$$

The matrix form of this operator is $T_{OAS} = B_{OAS}\mathscr{A}$, where $\mathscr{A}$ is the stiffness matrix and B_{OAS} is the Additive Schwarz preconditioner

$$B_{OAS} = R_0^T A_0^{-1} R_0 + \sum_{k=1}^{N}\sum_{l=1}^{M} R_{kl}^T A_{kl}^{-1} R_{kl}. \tag{15}$$

Here, R_{kl} are restriction matrices with $0,1$ entries returning the coefficients of the basis functions belonging to the local spaces V_{kl} and A_{kl} are the local stiffness ma-

trices restricted to the subspace V_{kl}. If the coarse space is nested into the fine space, R_0^T is the coarse-to-fine interpolation matrix and A_0 is the coarse stiffness matrix associated with the coarse space V_0. If the coarse space is non-nested, R_0^T is the coarse-to-fine L^2-projection matrix and the coarse space stiffness matrix is given by $A_0 = R_0 \mathscr{A} R_0^T$.

A convergence rate bound. Given the overlap index r defined before (12), we define the overlap parameter

$$\gamma = h(2r+2), \tag{16}$$

that is related to the width δ of the overlapping region by the bounds $\gamma = h(2r+2) \leq \delta \leq h(2r+p+1) \leq \frac{p+1}{2}\gamma$. Assuming that a) the parametric mesh is quasi-uniform, and b) the overlap index r is bounded from above by a fixed constant, we have the following result (see [4]).

Theorem 1. *The condition number of the 2-level additive Schwarz preconditioned operator T_{OAS} defined in (14), with either nested or non-nested coarse space, is bounded by*

$$\kappa_2(T_{OAS}) \leq C\left(1+\frac{H}{\gamma}\right),$$

where $\gamma = h(2r+2)$ is the overlap parameter defined in (16) and C is a constant independent of h, H, N, γ (but not of p, k).

4 Numerical results

In this section, we test the convergence properties of the isogeometric OAS preconditioner defined in (15) for linear elasticity problems on 3D domains. The IGA discretization with mesh size h, polynomial degree p, regularity k, is carried out by using the Matlab isogeometric library GeoPDEs [7]. The domain is decomposed into N overlapping subdomains of characteristic size H and overlap index r. The resulting linear system is solved by PCG with the isogeometric OAS preconditioner, with zero initial guess and a stopping criterion of 10^{-6} reduction of the relative residual.

Table 1 shows the scalability of the proposed isogeometric OAS preconditioner for a reference cubic domain decomposed into an increasing number of subdomainscho N of fixed subdomain size $H/h = 4$ (scaled speedup test), $p = 3$, $k = 2$, overlap $r = 0$ and $r = 1$, and both nested (left) and non-nested (right) coarse spaces. In addition to scalability, the results show that the two coarse spaces have similar performances and both improve when increasing the overlap size.

Table 2 illustrates the robustness of the OAS preconditioner for composite materials where the Young modulus E presents discontinuities across subdomain boundaries. The deformed 3D domain is a twisted bar shown in Fig. 1 (right), discretized by $16 \times 16 \times 8$ fine elements, $N = 4 \times 4 \times 2$ subdomains, and NURBS with $p = 3$ and $k = 2$ (except at the subdomain interfaces where $k = 0$). In the central jump test, the jump region consists of the $2 \times 2 \times 2$ central subdomains. Outside the jump

	nested coarse space				non-nested coarse space			
	$r=0$		$r=1$		$r=0$		$r=1$	
N	$\kappa_2=\lambda_{\max}/\lambda_{\min}$	n_{it}	$\kappa_2=\lambda_{\max}/\lambda_{\min}$	n_{it}	$\kappa_2=\lambda_{\max}/\lambda_{\min}$	n_{it}	cho $\kappa_2=\lambda_{\max}/\lambda_{\min}$	n_{it}
$2\times2\times2$	$16.3=8.03/0.49$	22	$9.1=8.25/0.91$	19	$17.2=8.03/0.47$	23	$9.3=8.25/0.89$	21
$3\times3\times3$	$18.5=8.04/0.43$	25	$11.2=9.31/0.83$	22	$22.8=8.04/0.35$	28	$12.8=9.68/0.76$	25
$4\times4\times4$	$19.8=8.04/0.41$	26	$11.9=9.47/0.80$	23	$20.1=8.04/0.40$	27	$12.0=9.47/0.79$	24
$5\times5\times5$	$20.2=8.04/0.40$	26	$12.1=9.52/0.79$	23	$20.5=8.04/0.39$	27	$12.4=9.53/0.77$	25
$6\times6\times6$	$20.4=8.05/0.40$	26	$12.3=9.56/0.78$	23	$20.6=8.05/0.39$	27	$12.5=9.56/0.76$	25

Table 1 Scalability of OAS preconditioner with nested (left) and non-nested (right) coarse space: condition number $\kappa_2(T_{OAS})$, extremal eigenvalues $\lambda_{\max}$, $\lambda_{\min}$ and PCG iteration counts n_{it} as a function of the number of subdomains N. Cubic domain, fixed $H/h=4$, $p=3$, $k=2$, $E=6e+6$, $\nu=0.3$.

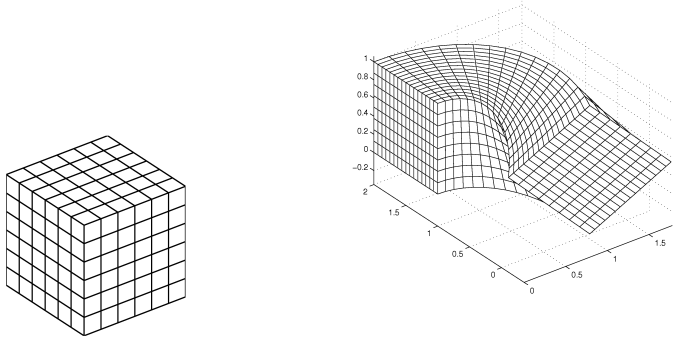


Fig. 1 3D domains used in the numerical tests.

region, $E=6e+3$ and $\nu=0.3$, while inside such region E has the value indicated in Table 2. In the checkerboard test, E alternates between the values $E=6e+3$ and $E=6e+7$, while $\nu=0.3$ everywhere. The results show that the unpreconditioned PCG deteriorate when E jumps towards $6e+7$, while the 2-level OAS preconditioner is very robust for jumps in E.

Jumping coefficient E, twisted quarter-ring domain								
		unpreconditioned		1-level OAS		2-level OAS		
	E	$\kappa_2 = \frac{\lambda_{\max}}{\lambda_{\min}}$	n_{it}	$\kappa_2 = \frac{\lambda_{\max}}{\lambda_{\min}}$	n_{it}	$\kappa_2 = \frac{\lambda_{\max}}{\lambda_{\min}}$	n_{it}
	$6e+1$	$1.01e+6 = \frac{8.48e+3}{8.40e-3}$	6029	$263.96 = \frac{8.00}{3.03e-2}$	57	$22.40 = \frac{8.47}{0.38}$	28
central	$6e+3$	$1.24e+4 = \frac{8.74e+3}{0.70}$	691	$261.04 = \frac{8.00}{3.06e-2}$	66	$25.79 = \frac{8.34}{0.32}$	30
jump	$6e+5$	$9.92e+5 = \frac{7.08e+5}{0.71}$	5793	$215.73 = \frac{8.00}{3.71e-2}$	55	$26.35 = \frac{8.58}{0.32}$	29
	$6e+7$	$7.85e+7 = \frac{7.08e+7}{0.90}$	20625	$191.83 = \frac{8.00}{4.17e-2}$	54	$30.93 = \frac{8.62}{0.28}$	30
checkerboard E		$8.10e+7 = \frac{3.29e+7}{0.41}$	20625	$70.32 = \frac{8.00}{0.11}$	32	$19.21 = \frac{8.50}{0.44}$	24

Table 2 OAS robustness with respect to jump discontinuities in E. Outside the central jump region of $2\times 2\times 2$ subdomains $E = 6e+3$ and $\nu = 0.3$. In the checkerboard test for E, $E = 6e+3$ or $E = 6e+7$. Condition number κ_2, extremal eigenvalues $\lambda_{\max}, \lambda_{\min}$ and iteration counts n_{it}. Fixed fine mesh $16\times 16\times 8$, $N = 4\times 4\times 2$, $H/h = 4$, $p = 3$, $k = 2$.

References

1. Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes, G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for *h*-refined meshes. *Math. Mod. Meth. Appl. Sci.*, **16**, 1–60, 2006.
2. L. Beirão da Veiga, D. Cho, L.F. Pavarino, S. Scacchi. Overlapping Schwarz methods for Isogeometric Analysis. *SIAM J. Numer. Anal.*, **50** (3), 1394-1416, 2012.
3. L. Beirão da Veiga, D. Cho, L.F. Pavarino, S. Scacchi. BDDC preconditioners for Isogeometric Analysis. Math. Mod. Meth. Appl. Sci. 23 (6): 1099–1142, 2013.
4. L. Beirão da Veiga, D. Cho, L.F. Pavarino, S. Scacchi. Isogeometric Schwarz preconditioners for linear elasticity systems. *Comp. Meth. Appl. Mech. Engrg.*, 253: 439–454, 2013.
5. M. Bercovier, I. Soloveichik. Additive Schwarz Decomposition methods applied to isogeometric analysis. Submitted for publication.
6. J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs. *Isogeometric Analysis. Towards integration of CAD and FEA*. Wiley, 2009.
7. C. De Falco, A. Reali, R. Vazquez. GeoPDEs: a research tool for Isogeometric Analysis of PDEs. *Advances in Engineering Software*, **42** (12), 1020-1034, 2011.
8. G.E. Farin. *NURBS curves and surfaces: from projective geometry to practical use*. A.K. Peters, 1995.
9. K. Gahalaut, J. Kraus, S. Tomar. Multigrid Methods for Isogeometric Discretization. *Comput. Methods Appl. Mech. Engrg.*, in press. doi: 10.1016/j.cma.2012.08.015.
10. T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Comp. Meth. Appl. Mech. Engrg.*, **194**, 4135–4195, 2005.
11. S. K. Kleiss, C. Pechstein, B. Jüttler, S. Tomar. IETI - Isogeometric Tearing and Interconnecting, *Comput. Methods Appl. Mech. Engrg.*, 247248: 201215, 2012.
12. B. F. Smith, P. Bjørstad, W. D. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
13. A. Toselli, O. B. Widlund. *Domain Decomposition Methods: Algorithms and Theory*. Computational Mathematics, Vol. 34. Springer-Verlag, Berlin, 2004.

Hybrid Domain Decomposition Solvers for the Helmholtz Equation

Martin Huber[1] and Joachim Schöberl[1]

1 Introduction

When solving the Helmholtz equation with standard finite elements, the oscillatory behavior of the solution results in a large number of degrees of freedom (DoFs) required to resolve the wave, especially for high wave numbers. This together with the indefiniteness of the problem makes an iterative solution of the resulting linear system of equations difficult. Nevertheless, some advances for finding efficient preconditioners for wave type problems have been made recently. Well known is the shifted Laplace Preconditioner [5], or a sweeping preconditioner [4] based on an approximate block $LDL^\top$ factorization, which is constructed layer by layer. Especially for parallel computing platforms domain decomposition methods are very popular. Apart from optimized Schwarz methods [8], i.e. Schwarz methods which rely on optimal transmission conditions as interface condition, the FETI-H [7] and the FETI-DPH [6] method are widely used. The last two methods can be seen as further developments of the FETI and the FETI-DP methods, respectively, specialized for Helmholtz problems.

The solution strategies presented in this work are based on a mixed hybrid discontinuous galerkin formulation [13, 10] Since the hybrid formulation provides appropriate interface conditions an efficient iterative solution with Krylov space methods combined with domain decomposition preconditioners is possible. Apart from adapting a BDDC preconditioner [3, 11] to the current setting, a new Robin type domain decomposition preconditioner is constructed. This preconditioner solves in each iteration step local problems on subdomains by directly inverting the subdomain matrix. Thus, it is well suited for parallel computations. Good convergence properties of both preconditioners are demonstrated by numerical experiments. The results of this paper will be presented in more detail in [9].

2 The Mixed Hybrid Discontinuous Galerkin Formulation

Our formulation is based on the mixed form of the Helmholtz equation: Find a scalar function $u : \Omega \to \mathbb{C}$ and a vector valued function $p : \Omega \to \mathbb{C}^d$

$$\operatorname{grad} u = \mathrm{i}\omega p \qquad \text{and} \qquad \operatorname{div} p = \mathrm{i}\omega u$$

[1]Institute for Analysis and Scientific Computing, Wiedner Hauptstrasse 8-10, A-1040 Wien, e-mail: {martin.huber}{joachim.schoeberl}@tuwien.ac.at

with an absorbing boundary condition $-p\cdot n+u=g$ on $\Gamma:=\partial\Omega$. As computational domain $\Omega\subset\mathbb{R}^d$ with $d=2,3$ a Lipschitz polyhedron is considered. Furthermore, n denotes the outer normal vector, the angular frequency ω is a positive constant and $g\in L^2(\Gamma)$. Note that [12] guarantees a unique solution.

In this paper we make use of the following notations. By $\mathscr{T}$ a triangulation with the elements T is denoted. The set of its facets F we call $\mathscr{F}$, n_F represents the normal vector onto a facet F, and n_T is the outer normal vector of an element T. Furthermore volume integrals are denoted by $\big(u,v\big)_T:=\int_T u\bar{v}\,\mathrm{d}x$ and surface integrals by $\big\langle u,v\big\rangle_{\partial T}:=\int_{\partial T}u\bar{v}\,\mathrm{d}s$.

In order to obtain efficient solvers for the Helmholtz equation, we consider it in a mixed hybrid form. Thus, we search for $(u,p,u_F,p_F)\in L^2(\Omega)\times H(\mathrm{div},\mathscr{T})\times L^2(\mathscr{F})\times L^2(\mathscr{F})=:U\times V\times U_F\times V_F$ such that for all $(v,q,v_F,q_F)\in U\times V\times U_F\times V_F$

$$\sum_{T\in\mathscr{T}}\Big(\big(i\omega u,v\big)_T-\big(\mathrm{div}p,v\big)_T-\big(u,\mathrm{div}q\big)_T-\big(i\omega p,q\big)_T+\big\langle u_F,n_T\cdot q\big\rangle_{\partial T}\\ +\big\langle n_T\cdot p,v_F\big\rangle_{\partial T}+\big\langle n_F\cdot p-p_F,n_F\cdot q-q_F\big\rangle_{\partial T}\Big)-\big\langle u_F,v_F\big\rangle_\Gamma=-\big\langle g,v_F\big\rangle_\Gamma. \tag{1}$$

This mixed hybrid formulation was introduced and discussed in [13]. In the formulation the space $H(\mathrm{div},\mathscr{T})$ represents an element wise $H(\mathrm{div})$ space without continuity constraints across element interfaces, and $L^2(\mathscr{F})$ is the space of L^2 functions on the facets. Consequently u_F and p_F are supported just on the facets, and they represent the values of u and $p\cdot n_F$ there. The problem is discretized by the finite dimensional spaces

$$U_h:=\prod_{T\in\mathscr{T}}P_k(T),\qquad V_h:=\prod_{T\in\mathscr{T}}RT_k(T),$$
$$U_{Fh}:=\prod_{F\in\mathscr{F}}P_k(F),\qquad V_{Fh}:=U_{Fh},$$

where polynomials of order k are denoted as P_k and RT_k represents a Raviart-Thomas element of order k. The discrete solutions we call u_h, p_h, u_{Fh} and v_{Fh}, respectively.

Since there is no global coupling for the functions u_h and p_h across different elements, the corresponding DoFs can be eliminated cheaply on the element level via static condensation [1]. Note that this elimination corresponds on each element to the solution of a wave type problem with Robin boundary conditions, and uniqueness is guaranteed. The resulting linear system of equations needs now to be solved only for the facet DoFs.

Remark 1. The original form of equation (1) in [13] contains a penalty parameter η, which was chosen to be one in our work. For this choice the local problem on the element, which needs to be solved during static condensation, corresponds to the original problem posed on the domain with $g=\pm p_F+u_F$. The sign depends on the direction of the facet normal n_F. Thus, g represents now the incoming impedance trace for the element.

In this work domain decomposition preconditioners will be used to solve the reduced linear system of equations for the facet unknowns u_{Fh} and p_{Fh}, which is obtained by eliminating the volume unknowns u_h and p_h. Note that this linear system of equations is related to the skeleton of a mesh, and a domain decomposition of the skeleton is induced by a decomposition of the underlying mesh. Since impedance traces are obtained from the facet unknowns by a simple transformation of variables, transmission conditions on the interface in the sense of [2] can be enforced by guaranteeing the same value of the facet unknowns of different subdomains on the subdomain interface. Thus the mixed hybrid formulation allows in a natural way for appropriate transmission conditions for domain decomposition preconditioners.

3 The BDDC preconditioner for the mixed hybrid formulation

In this section, we adapt the BDDC preconditioner introduced by Dohrmann in [3] (compare also [11]) to wave type problems. Therefore a stabilization term has to be added to the mixed hybrid formulation, more precisely, the term

$$\sum_{T\in\mathscr{T}} \gamma\Big(\big\langle (n_T\cdot n_F)p_F, v_F\big\rangle_{\partial T\setminus\Gamma} + \big\langle u_F, (n_T\cdot n_F)q_F\big\rangle_{\partial T\setminus\Gamma}\Big), \qquad \gamma\in\mathbb{C} \qquad (2)$$

is added to (1). The parameter $\gamma\in\mathbb{C}$ is a tuning parameter, we choose based on numerical experiments. For the Helmholtz and the vector valued wave equation we made good experience with $\gamma=-0.5-0.1i$. These additional terms are just added for inner facets, and because of the different sign of $n_T\cdot n_F$ for the two neighboring elements, they cancel out when the global system of equations is assembled. Thus the problem does not change. But for domain decomposition preconditioners, which are based on submatrices assembled just for a subdomain the situation changes. These additional terms do not cancel out in the submatrices for DoFs located on the interface to other subdomains.

We use a BDDC preconditioner for this modified facet problem. The computational domain is divided into subdomains, and the DoFs on facets which just belong to one subdomain are considered to be primal, as well as the low order DoFs on interface facets. The high order DoFs on interface facets are the dual ones.

This choice leads to a large global system for the primal DoFs. Note that this system of equations consists due to the missing high order unknowns at the interface of weakly coupled subdomain blocks. Therefore it can be solved rather efficiently by direct solvers on parallel computing platforms.

4 A Robin type domain decomposition preconditioner

Like the BDDC preconditioner, the new Robin type domain decomposition (RDD) preconditioner will be applied to the skeleton problem, including the stabilizing terms (2) with the same value for γ.

Before describing the preconditioner, some notations are required. We assume, that the computational domain is divided into N subdomains Ω_i. For each subdomain a matrix A_i representing the subdomain problem is subassembled, and the global matrix A of the linear system of equations is obtained by adding these submatrices. By $\tilde{A}_i$ we denote the block of A_i which corresponds to DoFs on inner facets, i.e. facets which just belong to the domain Ω_i. The matrix $R^{(i)}$ restricts a vector to the components corresponding to these inner DoFs of the domain Ω_i. The matrix $R_D^{(i)}$ provides a weighted restriction to the domain Ω_i, i.e. when applying it, a vector entry is divided by the number of subdomains to which the corresponding DoFs belongs to. Note that an application of the prolongation matrix $R_D^{(i)\top}$ results again in a division for the interface DoFs. Thus, by summing up over all subdomains, a mean value on the interface can be created.

Using this notations, a RDD step for finding $\tilde{x} := C_{RDD}^{-1} b$ with b as right hand side of the linear system of equations and C_{RDD} as the preconditioner reads as

1) $y_0 = 0$,
2) $y_1 = y_0 + \sum_{i=1}^{N} R^{(i)\top} \tilde{A}_i^{-1} R^{(i)} (b - A y_0)$,
3) $y_2 = y_1 + \sum_{i=1}^{N} R_D^{(i)\top} A_i^{-1} R_D^{(i)} (b - A y_1)$,
4) $\tilde{x} = y_2 + \sum_{i=1}^{N} R^{(i)\top} \tilde{A}_i^{-1} R^{(i)} (b - A y_2)$.

In step 2, the system of equations is solved exactly for the DoFs on the inner facets under the constraint that the solution on the interface is zero. Step 3 provides an update for the interface solution by partitioning the actual residual among the subdomains and solving the problem there exactly. A continuous interface solution is constructed by averaging the different subdomain solutions. Finally, in step 4 the solution is updated by solving the system of equations exactly for the DoFs on inner facets. Note that the interface solution remains unchanged.

The RDD-preconditioner can also be introduced in the variational projector notation. Therefore, we denote the bilinear form representing the Schur complement system, which is defined on the facet space $W := U_{Fh} \times V_{Fh}$ by a. Additionally, it is assumed that the bilinear form a can be decomposed into its subdomain contributions a_i, i.e. $a = \sum_{i=1}^{N} a_i$. The subspace of W containing the functions which are supported on the subdomain Ω_i is denoted by W_i, and in $\tilde{W}_i$ functions supported only on inner facets of the domain Ω_i are collected. The operator representation of the restriction matrix $R_D^{(i)}$ is called $\mathscr{R}_D^{(i)} : W \to W_i$. Thus, when applying it to any function in W, the function is restricted to the domain Ω_i, and its values on the interface facets are divided by the number of neighboring subdomains. Furthermore, $\mathscr{R}^{(i)} : W \to \tilde{W}_i$ is the restriction operator corresponding to the matrix $R^{(i)}$, and by $\mathscr{R}^{(i)\top}$ the prolongation operators are denoted.

Based on this, we define the variational projector $\mathscr{P}_D^{(i)}$ via $\mathscr{P}_D^{(i)} = \mathscr{R}_D^{(i)\top}\hat{\mathscr{P}}_D^{(i)}$ with the projector $\hat{\mathscr{P}}_D^{(i)} : W \to W_i$ and

$$a_i(\hat{\mathscr{P}}_D^{(i)}u, \phi) = a(u, \mathscr{R}_D^{(i)\top}\phi) \qquad \forall \phi \in W_i.$$

In the same way the variational projector $\mathscr{P}^{(i)}$ with $\mathscr{P}^{(i)} = \mathscr{R}^{(i)\top}\hat{\mathscr{P}}^{(i)}$ can be introduced. Here, $\hat{\mathscr{P}}^{(i)} : W \to \tilde{W}_i$ is given via

$$a_i(\hat{\mathscr{P}}^{(i)}u, \phi) = a(u, \mathscr{R}^{(i)\top}\phi) \qquad \forall \phi \in \tilde{W}_i.$$

If the operator $\mathscr{A}$ corresponds to the bilinear form a, and $\mathscr{I}$ is the identity, the error propagation operator $\mathscr{E}$ of the RDD-preconditioner reads as

$$\mathscr{E} = \mathscr{I} - \mathscr{C}_{RDD}^{-1}\mathscr{A} = \Big(\mathscr{I} - \sum_{i=1}^{N}\mathscr{P}^{(i)}\Big)\Big(\mathscr{I} - \sum_{i=1}^{N}\mathscr{P}_D^{(i)}\Big)\Big(\mathscr{I} - \sum_{i=1}^{N}\mathscr{P}^{(i)}\Big).$$

Remark 2. Because the system of equations is always solved exact for the DoFs on inner facets, both sets of facet DoFs u_{Fh} and p_{Fh} are not needed anymore, and the problem can be formulated just by using u_{Fh}. On the interface, both types of unknowns are still necessary in order to fix continuity conditions of the impedance traces across the interface and to guarantee convergence of the iterative solver. Nevertheless, neglecting one type of facet unknowns on inner facets saves many DoFs in an actual calculation.

5 Numerical Results

For all numerical examples which are presented in this section, we made good experience by taking a CG-solver, although, there exists no convergence theory for complex symmetric problems. We start the numerical results section by comparing the preconditioners for a simple two dimensional model problem. There, the Helmholtz equation is solved on a square $\Omega = [-1,1]^2$ with an incoming wave from above of Gaussian amplitude, fixed by the absorbing boundary condition. The computations were done with the MPI-parallel finite element code Netgen/Ngsolve (see *http://sourceforge.net/projects/ngsolve* or [14]), which contains the software package Metis for partitioning the domain. If not said differently, a Dell R-910 Server (4 Xeon E7 CPUs with 10 cores a 2.2 GHz, 512 GB RAM) was used.

In Table 1 the iteration numbers for the BDDC and the RDD preconditioner for different wavelengths $\lambda := \frac{2\pi}{\omega}$ and mesh sizes h are given. For all computations the polynomial order was kept constant to four, and nine subdomains were used for the preconditioners.

According to the table, the BDDC preconditioner shows the highest iteration numbers close to the resolution limit at $h \approx \lambda$, which corresponds for a polynomial order of $p = 4$ to about four unknowns per wavelength. When increasing the number

Table 1 Iteration numbers of the BDDC/RDD preconditioner using 9 subdomains ($p = 4$) for different mesh sizes and wavelength λ.

λ	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
$h = \frac{1}{4}$	45 / 78	49 / 65	60 / 61				
$h = \frac{1}{8}$	51 / 95	48 / 84	56 / 71	73 / 70			
$h = \frac{1}{16}$	56 / 123	50 / 96	49 / 83	59 / 73	80 / 72		
$h = \frac{1}{32}$	63 / 154	57 / 125	48 / 101	49 / 84	65 / 74	85 / 74	
$h = \frac{1}{64}$	66 / 202	62 / 164	56 / 127	50 / 111	50 / 89	74 / 82	101 / 89

of unknowns per wavelength, either by decreasing the mesh size or by increasing the wavelength, the number of iterations stays constant or grows slightly. For the RDD preconditioner the situation is vice versa. Although, for a large wavelength and a small mesh size the RDD preconditioner needs much more iterations than the BDDC preconditioner, it gets more and more competitive if the number of degrees of freedom per wavelength is reduced. Considering, that the RDD preconditioner is faster than the BDDC preconditioner with respect to setup-time and time per iteration, it is the method of choice for discretizations close to the resolution limit of the wave.

One reason for this behavior could be the different structure of the two solvers. While the RDD preconditioner allows just for local corrections, the BDDC solver benefits additionally from a coarse grid solution. For a decreasing wavelength, the solution gets more and more oscillatory, and the coarse grid correction, which provides communication across the whole subdomain loses its importance.

The number of iterations of the BDDC and the RDD preconditioner is also influenced by the size of the subdomains the computational domain is divided into. In Figure 1 iteration numbers of these two preconditioners are plotted for different wavelengths against the subdomain size H, both in logarithmic scale. Note that the partitioning of the domain was done by Metis, and therefore, H represents an average subdomain size. In the corresponding experiments the mesh size was kept constant to $\frac{1}{64}$ and the polynomial order to four. For the RDD preconditioner the number of iterations decreases with an increasing subdomain size. Figure 1 indicates that this decrease is proportional to $H^{-\alpha}$. According our experimental data α was estimated to be approximately 0.65. The situation is slightly different for the BDDC preconditioner. While it shows the same features for small wavelengths, i.e. for settings close to the resolution limit, the iterations stay almost constant for large wavelengths. A reason for this is, that for less oscillatory solutions the BDDC preconditioner benefits from its coarse grid correction.

Finally, we want to demonstrate the efficiency of our preconditioners with a three dimensional large scale example. The computational results presented in the following have been achieved using the Vienna Scientific Cluster 2 (VSC2). In this example, the solution of the Helmholtz equation for a grating (compare Figure 2) with period 0.14 was computed. The diameter of the computational domain was two. Thus, assuming a wave incoming from the top with Gaussian amplitude and wavelength

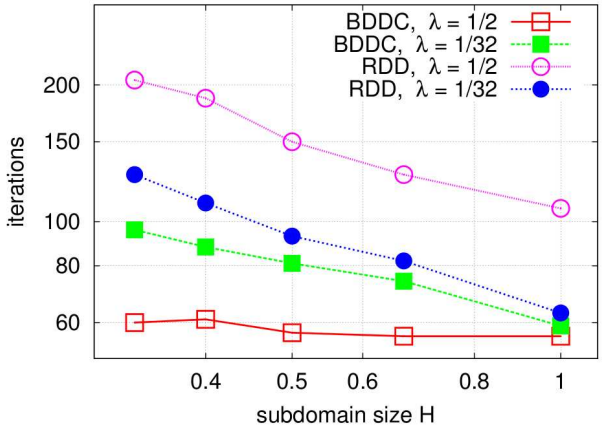


Fig. 1 Number of iterations plotted versus the size of one subdomain H for the BDDC and the RDD preconditioner. The polynomial order was 4 and $h = \frac{1}{64}$.

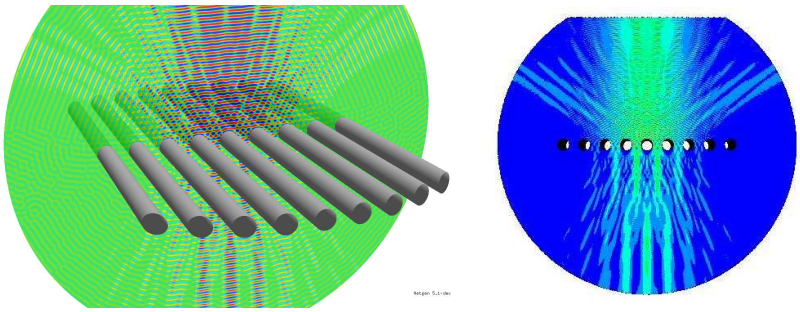


Fig. 2 Real part of the solution (left) and its absolute value (right) for a wave diffracted at a grating.

0.025 corresponds to an effective domain size of 80 wavelengths. For this setting, the left hand plot in Figure 2 shows the real part of the solution, and the absolute value is plotted on in the righthand plot. In the calculation, the underlying mesh had about 1.61 million elements with a maximal mesh size of 0.021. Selecting a polynomial order of $p = 4$ results in approximately 288.8 million volume unknowns (56.5 M. for u and 232.3 M. for p) and 98.0 million facet unknowns (49.0 M. for u_F and p_F). Using 1200 subdomains, the assembly of the matrix took 58 seconds and the setup of the RDD preconditioner 33 seconds. The problem was solved in 12.9 min-

utes with 399 iterations on 1200 processors. Recovering the volume DoFs u_h and p_h from the facet DoFs u_{Fh} and p_{Fh} took 53 seconds.

References

1. Arnold, D., Brezzi, F.: Mixed and nonconforming finite element methods: Implementation, postprocessing and error estimates. RAIRO Model. Math. Anal. Numer. **19**(1), 7–32 (1985)
2. Després, B.: Méthodes de décomposition de domains pour les problèms de propagation d'ondes en régime harmonique. Phd thesis, Université Paris IX Dauphine (1991)
3. Dohrmann, C.: A preconditioner for substructuring based on constrained energy minimization. SIAM J. Sci. Comput. **25**(1), 246–258 (2003)
4. Engquist, B., Ying, L.: Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation. Comm. Pure Appl. Math. **64**(5), 697–735 (2011)
5. Erlangga, Y.: Advances in iterative methods and preconditioners for the Helmholtz equation. Arch. Comput. Methods Eng. **15**(1), 37–66 (2008)
6. Farhat, C., Avery, P., Tezaur, R., Li, J.: FETI-DPH: a dual-primal domain decomposition method for acoustic scattering. J. Comput. Acoustics **13**(3), 499–524 (2005)
7. Farhat, C., Macedo, A., Lesoinne, M.: A two-level domain decomposition method for the iterative solution of high frequency exterior helmholtz problems. Numer. Math. **85**(2), 283–308 (2000)
8. Gander, M., Magoulès, F., Nataf, F.: Optimized Schwarz methods without overlap for the Helmholtz equation. SIAM J. Sci. Comput. **24**(1), 38–60 (2002)
9. Huber, M.: Hybrid discontinuous galerkin methods for the wave equation. Phd thesis, Vienna University of Technology `http://www.asc.tuwien.ac.at/~mhuber/thesis_huber.pdf`
10. Huber, M., Pechstein, A., Schöberl, J.: Hybrid domain decomposition solvers for the helmholtz and the time harmonic maxwell's equation. In: Domain Decomposition Methods in Science and Engineering XX, LNCSE. Springer, **91**, 279–287 (2013)
11. Li, J., Widlund, O.: FETI-DP, BDDC, and block Cholesky methods. Int. J. Numer. Meth. Engrg. **66**(2), 250–271 (2006)
12. Melenk, J.: On generalized finite element methods. Phd thesis, University of Maryland (1995)
13. Monk, P., Sinwel, A., Schöberl, J.: Hybridizing Raviart-Thomas elements for the Helmholtz equation. Electromagnetics **30**(1), 149–176 (2010)
14. Schöberl, J.: NETGEN - an advanced front 2D/3D-mesh generator based on abstract rules. Comput. Vis. Sci. **1**(1), 41–52 (1997)

Efficient implementation of a multi-level parallel in time algorithm

Matthew Emmett[1] and Michael L. Minion[2]

1 Introduction

The last decade has seen an increase in research into the parallelization of numerical methods for ordinary and partial differential equations in the temporal direction. One strategy for temporal parallelization involves decomposing the solution into time slices, which are distributed across processors or groups of processors, and employing an iterative scheme for computing the solution on all time slices in parallel [4, 3, 2]. The communication between time slices in these algorithms is quite regular, where each processor must send updates to the initial condition to the processor representing the following time slice. This communication must be done during each iteration of the method, and the amount of data sent is proportional to the size of the problem being solved. Although this communication takes place less frequently than that which typically occurs in spatially parallelized solvers for PDEs, the size of the data that must be transmitted is relatively large, and hence, reducing the effective cost of this data transfer is necessary to avoid reduced parallel efficiency.

In [2] a new approach for the temporal parallelization of the numerical solution to partial differential equations, called the Parallel Full Approximation Scheme in Space and Time (PFASST), is introduced. PFASST is similar in structure to the earlier Parareal [4] and PITA [3] methods, but uses a deferred correction type procedure first described in [5, 7] within time slices instead of a traditional direct method, which provides an improved theoretical maximum parallel efficiency as compared to Parareal or PITA. The PFASST algorithm also uses a hierarchy of spatial and temporal discretizations of the problem, wherein coarse problems are defined in a procedure analogous to the full approximation scheme (FAS) used extensively in multigrid methods for nonlinear problems (see e.g., [1]). Since FAS is naturally recursive, an extension of the approach in [2] to multiple levels of spatial and temporal refinement is possible. The key algorithmic change in PFASST presented here concerns the issue of the communication cost.

The PFASST method is reviewed here in Sect. 2. In Section 3, an approach is outlined wherein corrections computed at different refinement levels are passed between processors in a way which can greatly reduce the communication overhead of the PFASST iterations. The timing results presented in Sect. 4 demonstrate the effectiveness of the proposed communication strategy. Finally, a short discussion of the current results and future research directions can be found in Sect. 5.

[1] Lawrence Berkeley National Laboratory, e-mail: mwemmett@lbl.gov .[2] Institute for Computational and Mathematical Engineering, Stanford University, e-mail: mlminion@gmail.com

2 PFASST

In this section, a brief description of the PFASST algorithm is included. It is assumed that the reader is familiar with Spectral Deferred Correction (SDC) methods and full approximation scheme (FAS) corrections. For more complete details, see [7, 2].

For the following description, consider the ODE initial value problem

$$u'(t) = f(t,u(t)), \qquad u(0) = u_0, \tag{1}$$

where $t \in [0,T]$; $u_0, u(t) \in \mathbb{C}^N$; and $f : \mathbb{R} \times \mathbb{C}^N \to \mathbb{C}^N$. It is assumed here that (1) represents a method of lines discretization of a PDE.

For a PFASST computation with L levels of spatial and temporal resolution (with level 0 being the finest), the time interval of interest $[0,T]$ is divided into N uniform intervals $[t_n, t_{n+1}]$ which are assigned to the processors P_n where $n = 0 \ldots N-1$. Each interval is subdivided on each level ℓ by defining $M_\ell + 1$ SDC nodes $t_\ell = [t_{\ell,0} \cdots t_{\ell,M_\ell}]$ such that $t_n = t_{\ell,0} < \cdots < t_{\ell,M_\ell} = t_{n+1}$, where we have omitted the dependence of t_ℓ on n for brevity. The SDC nodes $t_{\ell+1}$ on level $\ell+1$ are chosen to be a subset of the SDC nodes t_ℓ on level ℓ to facilitate interpolation and restriction between coarse and fine levels. Note that the use of point injection as the coarsening procedure with Gaussian quadrature nodes means that the coarse nodes may not correspond to Gaussian nodes. The solution at the m^{th} node on level ℓ during iteration k is denoted $U(\ell,k,m)$. For brevity let $U(\ell,k) = [U(\ell,k,0), \cdots, U(\ell,k,M_\ell)]$ and $F(\ell,k) = [F(\ell,k,0), \cdots, F(\ell,k,M_\ell)] = [f(t_{\ell,0}, U(\ell,k,0)), \cdots, f(t_{\ell,M_\ell}, U(\ell,k,M_\ell))]$.

In the parareal method, the processors are typically initialized by using the coarse propagator in serial to yield a low-accuracy initial condition for each processor. In [2], an alternative initialization scheme is described. During initialization, each processor begins coarse SDC sweeps immediately using the initial condition from the first processor. Hence the number of coarse iterations (SDC sweeps) done on processor P_n in the initialization is equal to n rather than 1. This has the same total computational cost of doing one coarse SDC sweep per processor in serial, but the additional SDC sweeps can improve the accuracy of the solution significantly, as is demonstrated in [2]. During this initial iteration, no communication is necessary since each processor computes the same data as the processor corresponding to the previous process. Hence further discussion of the initialization procedure is omitted.

The full PFASST iterations for $k = 0 \ldots K-1$ on each processor P_n proceed as follows. Assuming that the fine solution and function values $U(0,k)$ and $F(0,k)$ are available, the iterations are comprised of the following steps:

(i) Perform one fine SDC sweep using the values $U(0,k)$ and $F(0,k)$. This will yield provisional updated values $U(0,k+1)$ and $F(0,k+1)$.
(ii) Send $U(0,k+1,M_0)$ to processor P_{n+1} if $n < N-1$. This will be received as the new initial condition $U(0,k+1,0)$ in the next iteration.
(iii) Go down the V-cycle: for each $\ell = 1 \ldots L-2$

a. Restrict the fine values $U(\ell-1,k+1)$ to the coarse values $U(\ell,k)$ and compute $F(\ell,k)$.
b. Compute the FAS correction $B(\ell,k)$ using $F(\ell-1,k+1)$, $F(\ell,k)$, and $B(\ell-1,k)$.
c. Perform n_ℓ SDC sweeps with the values $U(\ell,k)$, $F(\ell,k)$ and the FAS correction $B(\ell,k)$. This will yield new values $U(\ell,k+1)$ and $F(\ell,k+1)$.
d. Send $U(\ell,k+1,M_\ell)$ to processor P_{n+1} if $n < N-1$. This will be received as the new initial condition $U(\ell,k+1,0)$ in the next iteration.

(iv) Perform the bottom sweep:

a. Restrict the fine values $U(L-2,k+1)$ to the coarse values $U(L-1,k)$ and compute $F(L-1,k)$.
b. Compute the FAS correction $B(L-1,k)$ using $F(L-2,k+1)$, $F(L-1,k)$, and $B(L-2,k)$.
c. Receive the new initial value $U(L-1,k,0)$ from processor P_{n-1} if $n > 0$ and compute $F(L-1,k,0)$.
d. Perform n_{L-1} coarse SDC sweeps using the values $U(L-1,k)$, $F(L-1,k)$ and the FAS correction $B(L-1,k)$. This will yield new values $U(L-1,k+1)$ and $F(L-1,k+1)$.
e. Send $U(L-1,k+1,M_{L-1)}$ to processor P_{n+1} if $n < N-1$. This will be received as the new initial condition $U(L-1,k,0)$ in the current iteration on the next processor P_{n+1}.

(v) Return up the V-cycle: for each $\ell = L-2\ldots 1$:

a. Interpolate coarse correction $U(\ell+1,k+1)-U(\ell+1,k)$ in space and time and add to $U(\ell,k+1)$. Recompute new values $F(\ell,k+1)$.
b. Receive the new initial value $U(\ell,k+1,0)$ from processor P_{n-1} if $n > 0$.
c. Interpolate correction $U(\ell+1,k+1,0)-U(\ell+1,k,0)$ to new $U(\ell,k+1,0)$ and recompute $F(\ell,k+1,0)$.
d. Perform n_ℓ SDC sweeps with the values $U(\ell,k+1)$, $F(\ell,k+1)$ and the FAS correction $B(\ell,k)$. This will once again yield new values $U(\ell,k+1)$ and $F(\ell,k+1)$.

(vi) Interpolate coarse correction $U(1,k+1)-U(1,k)$ in space and time and add to $U(0,k+1)$. Recompute new values $F(0,k+1)$.
(vii) Receive the new initial value $U(0,k+1,0)$ from processor P_{n-1} if $n > 0$.
(viii) Interpolate correction $U(1,k+1,0)-U(1,k,0)$ to new $U(0,k+1,0)$ and recompute $F(0,k+1,0)$.

The steps above are illustrated in Figure 1(b), in which solid blocks denote SDC sweeps (F_ℓ) and gradient blocks denote interpolation ($I^\ell_{\ell+1}$) or restriction ($R^{\ell+1}_\ell$). The length of the blocks are proportional to their cost, with fine SDC sweeps being 4 and 16 times more expensive than intermediate and coarse SDC sweeps, respectively (which would correspond to a 1D PFASST scheme with both spatial and temporal refinements by a factor of 2). The length of the interpolation and restriction blocks is

also proportional to their cost: when transferring between levels we must re-evaluate the function values $F(\ell,k)$ in order to compute the FAS corrections $B(\ell,k)$.

3 Communication between processors

In the precursors to this work appearing in [7, 2] as well as the original papers on the parareal method, little attention is given to the topic of scheduling the communication between processors. In this section, a strategy which effectively unblocks communication except at the coarsest resolution is presented. It is assumed here that the parallel implementation of PFASST allows computation and communication to be performed simultaneously.

In each PFASST iteration, the full solution (or correction to the solution) must be passed forward in time to the next processor. In the two level scheme presented in [7, 2], this is done directly after the coarse correction is applied to the current fine solution. Since the coarse SDC sweep on the next processor cannot begin until this data is received, scheduling the communication in this way results in a *blocking* communication. The blocking communication is depicted in Fig. 1(a), where each column represents the operations done on a processor with wall time progressing from bottom to top. The white and black circles correspond to the send and receive process on each processor. After the coarsest SDC sweep (denoted $\mathscr{F}_1$), the full update of the initial condition is sent forward in time. The white gap represents the waiting time, which grows linearly with the number of processors.

Note in Fig. 1(a), the first operation performed after a processor receives data is a coarse SDC sweep. In order to perform this sweep, a new initial condition is required, but only at the coarse resolution. The key observation used here is that it is only necessary to pass the corrections to the initial data during each PFASST iteration, and more importantly this communication can be decomposed into corrections corresponding to each level of spatial resolution. Although this means that more data in total is being passed during each PFASST iteration, data from the finer levels can be sent before the corresponding fine SDC sweeps are performed on each processor. Therefore, if the computational cost of the computation at the coarser levels is greater than the communication cost of sending data at a particular level, then the communication becomes *non-blocking*.

For example, consider Fig. 1(b), which diagrams the scheduling of communication for a three-level implementation of the PFASST algorithm. At each level, as soon as an SDC sweep is completed (denoted by $\mathscr{F}_\ell$ for $\ell = 0 \ldots 2$), the correction to the solution at the final SDC node (which corresponds to the first SDC node on the next processor) is sent. This can be done before the recursive call to compute a correction at the next coarsest level (denoted by the blocks $R_\ell^{\ell+1}$). The sent data can then be received in a buffer at the next processor and is not needed until after the corresponding coarse correction has been computed (denoted by the blocks $I_{\ell+1}^{\ell}$) on that processor. Hence, the sending of the finest data overlaps with the computation of the correction on two coarser levels. It is only at the coarsest level that there is no

computational work to be done while waiting for the data to be received. However, if the coarse data is significantly smaller than fine data, the communication cost at the coarsest level is likewise significantly reduced. In the three-level, three-dimensional example in Sect. 4, the coarsest level contains 1/64 the amount of data as the finest level with communication time similarly reduced.

It should be noted that the crossing of the lines corresponding to communication in Fig. 1(b) assume that blocking coarse communication could be scheduled to interrupt non-blocking fine level communication, a feature which may not exist in a standard message passing library. If this is not the case, there is still the opportunity to overlap computation with communication before the blocking coarsest level send occurs. Finally, recall that the work performed by each processor in Fig. 1(b) is not uniform since processor P_n does n coarse SDC sweeps during the initialization procedure.

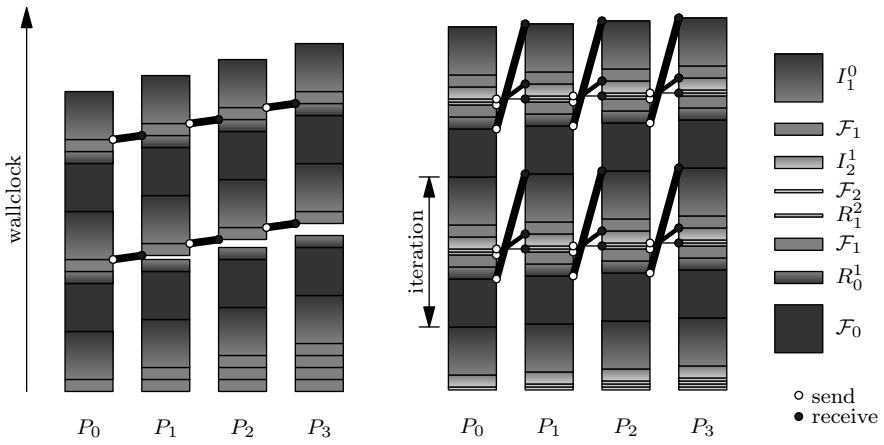


Fig. 1 Left: (a) Communication diagram for the original 2-level PFASST algorithm. Right: (b) Communication diagram for the 3-level V-cycle PFASST algorithm.

4 Timing

Timing information for a three-level PFASST run was obtained for a three dimensional model problem: the incompressible Navier-Stokes equations given by

$$u_t + u \cdot \nabla u = \nu \nabla^2 u - \nabla p \nabla \cdot u = 0. \tag{2}$$

A method of lines approach is employed by placing the equations in projection form and using spectral approximations to all spatial derivatives via the FFT [6]. The advective piece of the equation is treated explicitly while the diffusive piece is treated

implicitly. The fine spatial discretization consists of 256^3 points in a unit cube, resulting in a total of 3×256^3 degrees of freedom on the fine level, or approximately 384 megabytes using 64 bits per degree of freedom. The fine temporal discretization consists of 5 Gauss-Lobatto SDC nodes. The run was performed across 16 processors of "Edison", the Cray XC30 system at the National Energy Research Scientific Computing Center (NERSC).

Figs. 2 and 3 present timing information for various parts of the PFASST algorithm across the processors for each PFASST iteration. From Fig. 2 we note that the iteration time (which encompasses all overhead costs including interpolation, restriction, and FAS computation) is fairly consistent across each processor and iteration, and that the cost of the intermediate and coarse sweeps are significantly cheaper than the fine sweep.

From Fig. 3 we note that the (blocking) coarse send and receive times are fairly significant (send/receive 0) between some processors. This establishes that communication across compute nodes is non-trivial even at the coarse level (recall that the coarse level consists of 3×64^3 degrees of freedom, which is 64 times less than the fine level). Finally, the fine and intermediate send and receive times (send/receive 1 and 2) are essentially zero across all processors and iterations. This demonstrates that the fine and intermediate communications are essentially non-blocking and were successfully overlapped with computation.

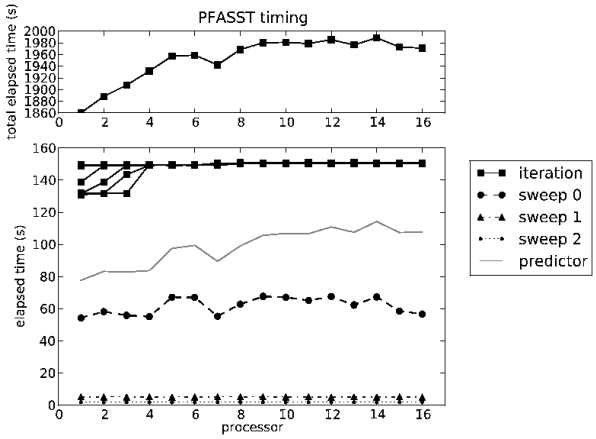


Fig. 2 Timing information for the three-dimensional Navier-Stokes solver. The top panel shows the total elapsed run time. The bottom panel shows iteration time (including all overhead), SDC sweep time for each level, and the initialization time (predictor).

The three-level PFASST run using 16 time processors desribed above achieves a speedup of roughly 7.2 compared to a serial SDC-based run (which requires 8 serial iterations per time step to acheive the same accuracy as 6 PFASST iterations). This corresponds to a parallel efficiency of roughly 45%. The parallel efficiency of

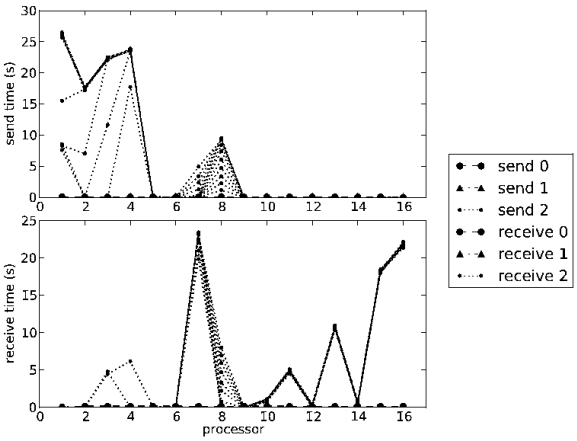


Fig. 3 Communication timing information for the three-dimensional Navier-Stokes solver. The top panel shows send time for each level, and the bottom panel shows receive time for each level. Note that the send and receive times for the intermediate and fine levels (1 and 0) are negligible compared to the coarse level (2).

PFASST can vary substantially depending on the number of processors, the error tolerance, and the sensitivity of the problem at hand [2].

5 Discussion

In summary, we have demonstrated how the necessary transfer of relatively large amounts of data between processors in the PFASST algorithm can be scheduled so that only a small amount of the transfer is blocking. As long as the computation involved in a recursive call to a coarser level correction is more expensive than the communication, the communication cost is negligible. The effectiveness of the scheduling procedure relies on the communication and computation being done simultaneously, and is optimal if blocking communication can interrupt non-blocking communication between two processors.

Current trends in the design of the next generation of large parallel computers suggest that the relative cost of data transfer between processors will continue to grow. In this case, more elaborate strategies to avoid blocking communication in the PFASST algorithm might become necessary. For example, since only the correction to the solution needs to be passed between processors, it is possible that fewer significant digits could be used to transmit data. The main point we stress here is that, except at the coarsest level, there is useful work that a processor can perform while data is being passed from processor to processor. In fact, the algorithm could be

reconfigured so that at each stage of the FAS procedure, SDC sweeps are performed at each level until the necessary data at the next finest level is received.

References

1. Briggs, W.L., Henson, V.E., McCormick, S.F.: A Multigrid Tutorial, vol. 72. SIAM (2000)
2. Emmett, M., Minion, M.: Toward an efficient parallel in time method for partial differential equations. Communications in Applied Mathematics and Computational Science **7**(1), 105–132 (2012)
3. Farhat, C., Chandesris, M.: Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications. Internat. J. Numer. Methods Engrg. **58**(9), 1397–1434 (2003)
4. Lions, J., Maday, Y., Turinici, G.: A "parareal" in time discretization of PDE's. Comptes Rendus de l'Academie des Sciences Series I Mathematics **332**(7), 661–668 (2001)
5. Minion, M., Williams, S.: Parareal and spectral deferred corrections. In: AIP Conference Proceedings, vol. 1048, pp. 388–391. AIP (2008)
6. Minion, M.L.: Semi-implicit projection methods for incompressible flow based on spectral deferred corrections. Appl. Numer. Math. **48**(3-4), 369–387 (2004)
7. Minion, M.L.: A hybrid parareal spectral deferred corrections method. Comm. Appl. Math. and Comp. Sci. **5**(2), 265–301 (2010)

Optimized Schwarz Methods and model adaptivity in electrocardiology simulations

Luca Gerardo-Giorda[1], Lucia Mirabella[2], and Mauro Perego[3] and Alessandro Veneziani[4]

1 Numerical Models for the Cardiac Potential

At the macroscopic level, the myocardial tissue can be regarded as the superposition of two continuous and anisotropic media, the intra-cellular and the extra-cellular one. They coexist and are connected by a cell membrane, whose capacitance is denoted by C_m. The tissue conductivity depends upon its cells orientation, and in the most general case the associated tensor is anisotropic [7, 14]. In any point $x \in \Omega$, where Ω is the spatial domain under consideration, it is possible to identify an orthonormal triplet of directions, $a_l(x)$, $a_t(x)$, $a_n(x)$, with $a_l(x)$ parallel to the fibers direction, and we denote by σ_τ^l, σ_τ^t, and σ_τ^n ($\tau = i,e$) the corresponding intra and extracellular conductivity coefficients. The conductivity tensors are given by

$$\mathbf{D}_\tau(x) = \sigma_\tau^l(x) a_l(x) a_l^T(x) + \sigma_\tau^t(x) a_t(x) a_t^T(x) + \sigma_\tau^n(x) a_n(x) a_n^T(x), \qquad \tau = i,e. \quad (1)$$

We assume that $\mathbf{D}_\tau$ fulfill in Ω a uniform elliptic condition.

The Bidomain model. The Bidomain model is a nonlinear reaction-diffusion system of parabolic type describing the spatio-temporal dynamics of the *intra* and *extracellular* potentials, denoted by u_i and u_e, while the cell membrane is regarded as dislocated in the domain [2]. We rely in this paper on a non-symmetric formulation in terms of the transmembrane potential $u = u_i - u_e$, and the extracellular one [4]. We denote by $\mathbf{u} = (u, u_e)^T$ the unknown, by $V = H^1(\Omega)\backslash\{c : c \in \mathbb{R}\}$ and by letting

$$\mathbf{D} = \begin{bmatrix} \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l + \sigma_e^l} & \frac{\sigma_e^l \mathbf{D}_i - \sigma_i^l \mathbf{D}_e}{\sigma_i^l + \sigma_e^l} \\ \mathbf{D}_i & \mathbf{D}_i + \mathbf{D}_e \end{bmatrix} \qquad \mathbf{E}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

the Bidomain system reads as follows. Find $\mathbf{u} \in L^2(0,T; H^1(\Omega) \times V)$, such that

$$\chi C_m \mathbf{E}_1 \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \mathbf{D} \nabla \mathbf{u} + \chi I_{ion}(u) \mathbf{e}_1 = \mathbf{I}^{app}, \quad (2)$$

where χ is the membrane area per tissue volume ratio, $I_{ion}(u)$ is a nonlinear function of the transmembrane potential u, specified by a ionic model, and where $\mathbf{I}^{app}$

[1] BCAM - Basque Center for Applied Mathematics, Bilbao, Spain, e-mail: lgerardo@bcamath.org [2] "W.H. Coulter" Dept. Biomed. Engrg., GaTech, Atlanta, GA, USA, e-mail: lucia.mirabella@bme.gatech.edu [3] Dept. of Sci. Comp., Florida State University, Thallahassee, FL, USA, e-mail: mperego@fsu.edu [4] Dept. of Math. and CS, Emory University, Atlanta, GA, USA, e-mail: ale@mathcs.emory.edu

represent the applied current stimuli. Several ionic models are available in literature, from more phenomenological to more accurate ones, but the choice of the nonlinear term $I_{ion}(u)$ does not have any influence on the procedure highlighted in what follows. The problem is completed by suitable initial conditions, and by homogeneous Neumann boundary conditions on $\partial\Omega$, modeling an insulated myocardium. The transmembrane potential u is uniquely determined from (2), while the extracellular potential u_e is determined up to a function of time, and is usually identified by imposing a zero average at each time ($\int_\Omega u_e(x,t)\,dx = 0$, for all $t \in (0,T)$).

The Monodomain model. The Monodomain model is a simplified model for the propagation of the electrical stimulus, based upon a proportionality assumption between the intracellular and the extracellular conductivity tensors, namely assuming $\mathbf{D}_e = \lambda \mathbf{D}_i$, where λ is a constant to be properly chosen. We assume here $\lambda = \sigma_e^l/\sigma_i^l$ [6], and the Monodomain model reads as follows. Find $u \in L^2(0,T;H^1(\Omega))$, such that

$$\chi C_m \frac{\partial u}{\partial t} - \nabla \cdot \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l + \sigma_e^l} \nabla u + \chi I_{ion}(u) = I^{app}. \tag{3}$$

Also system (3) is coupled with suitable initial conditions, and homogeneous Neumann boundary conditions on $\partial\Omega$. Differently from the Bidomain, the Monodomain model features a unique solution and is cheaper to solve numerically. In absence of applied currents, the Monodomain model is accurate enough to catch the desired dynamics and effects of the action potential propagation [12]. However, the Bidomain model becomes necessary when current stimuli are applied in the extracellular space. Also, the Monodomain is inadequate to simulate defibrillation [16].

1.1 Numerical approximation

Time integration. For simplicity in presentation, we consider a fixed time step Δt, and we denote with superscript n the unknowns computed at time $t^n = n\Delta t$. Both the Bidomain (2) and the Monodomain equations (3) are advanced in $(0,T)$ by a semi-implicit scheme, where the nonlinear term (the ionic current) is evaluated at the previous time step [2, 4]. More precisely, moving from t^n to t^{n+1} we solve in Ω

$$\chi C_m \mathbf{E}_1 \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} - \nabla \cdot \mathbf{D} \nabla \mathbf{u}^{n+1} = \mathbf{I}^{app} - \chi I_{ion}(u^n)\,\mathbf{e}_1 \tag{4}$$

for the Bidomain system, and

$$\chi C_m \frac{u^{n+1} - u^n}{\Delta t} - \nabla \cdot \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l + \sigma_e^l} \nabla u^{n+1} = \mathbf{I}^{app} - \chi I_{ion}(u^n) \tag{5}$$

for the Monodomain one.

Space discretization. Both Bidomain (4) and Monodomain (5) models are discretized in space by finite elements [2, 8, 15]. When solving the Bidomain system, the unknowns of the fully discrete problem are represented by the vector $\left(u_h, u_{e,h}\right)^T$, storing the nodal values of the transmembrane and extracellular potentials. The matrix associated with the discrete Bidomain models is given by

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{uu} & \mathbf{B}_{ue} \\ \mathbf{B}_{eu} & \mathbf{B}_{ee} \end{bmatrix} = \begin{bmatrix} \frac{\chi C_m}{\Delta t}\mathbf{M} + \frac{\sigma_e^l}{\sigma_i^l+\sigma_e^l}\mathbf{K}_i & \frac{\sigma_e^l}{\sigma_i^l+\sigma_e^l}\mathbf{K}_i - \frac{\sigma_i^l}{\sigma_i^l+\sigma_e^l}\mathbf{K}_e \\ \mathbf{K}_i & \mathbf{K}_i + \mathbf{K}_e \end{bmatrix}, \qquad (6)$$

where $\mathbf{M}$ is the mass matrix while $\mathbf{K}_i$ and $\mathbf{K}_e$ are the stiffness matrices associated with the chosen finite elements space.
When solving the Monodomain system, the unknown of the fully discrete problem is u_h, and the associated matrix is simply block $\mathbf{B}_{uu}$ of the matrix $\mathbf{B}$ in (6).

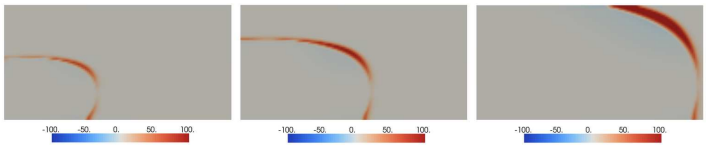


Fig. 1 Differences in the propagation of the membrane potential between Bidomain (u_{Bido}) and Monodomain (u_{Mono}) simulation: $u_{Bido} - u_{Mono}$, with fibers oriented along the x axis (from [6]).

2 A model adaptive strategy

In Fig. 1 (from [6]) we report the differences of the transmembrane potential computed with the Bi- and Monodomain models respectively at different instants. The Figure pinpoints that the differences are mainly concentrated around the wavefront. From these results, we argue that the Monodomain provides an accurate approximation of the potential in most of the region of interest. The *model adaptive strategy* consists then in solving the Bidomain only when actually needed. In a first implementation of this approach [9] a suitable *a posteriori* model estimator was introduced. A hybrid model called *Hybridomain* was advocated. The latter assembles the block $\mathbf{B}_{ue}$ only in correspondence with the nodes identified as Bidomain ones by the model estimator, while the second equation stays untouched. This simplifies significantly the implementation, however the computational advantage is limited, since also in the Monodomain regions an extended problem with the same size of the Bidomain one is solved. An alternative procedure consists of a genuine heterogeneous coupling by splitting the domains where the two models are solved. This coupling raises non trivial issues when matching the two models, featuring a different size. This has been considered in [6], where the Optimized Schwarz method

has been advocated for the heterogeneous coupling, addressing the matching conditions at the interface between two different domains. Here, we focus on practical issues when using this approach in realistic problems. A first idea would be to trivially use the *a posteriori* error estimator for detecting the regions where to solve the Bidomain problem and then to couple these subdomains with the Monodomain regions. However, this approach is barely doable. As a matter of fact. the Robin-type interface conditions in the Optimized Schwarz setting require the assembly of mass matrices on the interfaces. As a consequence, every time the Bidomain region changes, one should identify the new interfaces and then recompute the matrices, with an additional computational cost that is anticipated to reduce the advantage of the Optimized Schwarz coupling. The model adaptive strategy we propose here relies instead on a *a priori* subdivision of Ω into smaller subdomains Ω_j. The model error estimator will associate runtime each subdomain with either the Bidomain or the Monodomain problem. In this way, the interfaces matrices needed for the coupling can be computed once at the beginning of the time loop. Notice that the non-symmetric formulation of the Bidomain system ensures that the matrices for the Monodomain model are available after assembling the Bidomain ones.

2.1 Coupling conditions and Optimized Schwarz Methods

We outline here the coupling conditions for the three different types of interfaces. If the subdomains involved have the same characteristic (Bido/Bido and Mono/Mono) the corresponding solutions are labeled by subscript 1 and 2, while if the subdomains have different characteristics (Bido/Mono) the corresponding solutions are labeled with subscript B and M.

Bidomain/Bidomain interface. The coupling conditions on the Bidomain/Bidomain interface have been introduced in [5], and are given by

$$\begin{aligned} &\mathbf{n}_1^T \mathbf{D}\nabla \mathbf{u}_1 + \alpha_1 \Sigma \mathbf{u}_1 = \mathbf{n}_1^T \mathbf{D}\nabla \mathbf{u}_2 + \alpha_1 \Sigma \mathbf{u}_2 \\ &\mathbf{n}_2^T \mathbf{D}\nabla \mathbf{u}_2 + \alpha_2 \Sigma \mathbf{u}_2 = \mathbf{n}_2^T \mathbf{D}\nabla \mathbf{u}_1 + \alpha_2 \Sigma \mathbf{u}_1, \quad \text{where} \quad \Sigma = \begin{bmatrix} \frac{\sigma_e^l}{\sigma_i^l+\sigma_e^l} & 0 \\ 1 & \frac{\sigma_i^l+\sigma_e^l}{\sigma_i^l} \end{bmatrix}. \end{aligned} \tag{7}$$

The convergence of the Optimized Schwarz Algorithm based on the interface conditions (7) was analyzed in [5], where also optimal parameters have been identified by means of Fourier analysis.

Bidomain/Monodomain interface. Due to a dimensional mismatch between the two models, two interface conditions are needed on the Bidomain side of the interface, and one on the Monodomain side [6]. Possible coupling conditions are

$$\begin{aligned}
&\mathbf{n}_B^T \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l+\sigma_e^l}(\nabla u_B + \nabla u_{e,B}) - \mathbf{n}_B^T \frac{\sigma_i^l \mathbf{D}_e}{\sigma_i^l+\sigma_e^l}\nabla u_{e,B} + \frac{\sigma_e^l \alpha}{\sigma_i^l+\sigma_e^l} u_B = \mathbf{n}_B^T \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l+\sigma_e^l}\nabla u_M + \frac{\sigma_e^l \alpha}{\sigma_i^l+\sigma_e^l} u_B \\
&\mathbf{n}_B^T \mathbf{D}_i(\nabla u_B + \nabla u_{e,B}) + \mathbf{n}_B^T \mathbf{D}_e \nabla u_{e,B} + \alpha u_B + \frac{\sigma_i^l+\sigma_e^l}{\sigma_i^l}\alpha u_{e,B} = \alpha u^{rest}
\end{aligned} \tag{8}$$

for the Bidomain subproblem, and

$$\mathbf{n}_M^T \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l+\sigma_e^l}\nabla u_M + \frac{\sigma_e^l \alpha}{\sigma_i^l+\sigma_e^l} u_M = \mathbf{n}_M^T \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l+\sigma_e^l}(\nabla u_B + \nabla u_{e,B}) - \mathbf{n}_M^T \frac{\sigma_i^l \mathbf{D}_e}{\sigma_i^l+\sigma_e^l}\nabla u_{e,B} + \frac{\sigma_e^l \alpha}{\sigma_i^l+\sigma_e^l} u_B \tag{9}$$

for the Monodomain one. To cope with the mismatch, the second condition in (8) is a transparent boundary condition, designed to avoid spurious reflexions off the interface for the extracellular potential wave. The convergence of the Optimized Schwarz Algorithm based on the interface conditions (8)-(9) was analyzed in [6], where also optimal parameters has been identified by means of Fourier analysis.

Monodomain/Monodomain interface. The Optimized Schwarz coupling is significantly simpler on the interface between two Monodomain regions. The semi-implicit temporal integration scheme reduces the problem at each time step to a linear steady reaction-diffusion problem, whose solution by means of Optimized Schwarz Methods has been widely studied, and an optimal parameter has been identified [3]. The coupling on the interface is given by

$$\begin{aligned}
&\mathbf{n}_1^T \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l+\sigma_e^l}\nabla u_1 + \alpha^{opt} u_1 = \mathbf{n}_1^T \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l+\sigma_e^l}\nabla u_2^p + \alpha^{opt} u_2^p \\
&\mathbf{n}_2^T \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l+\sigma_e^l}\nabla u_2 + \alpha^{opt} u_2 = \mathbf{n}_2^T \frac{\sigma_e^l \mathbf{D}_i}{\sigma_i^l+\sigma_e^l}\nabla u_1 + \alpha^{opt} u_1.
\end{aligned} \tag{10}$$

2.2 The model error estimator

The *a posteriori* error estimator for choosing between a Bidomain or Monodomain simulation in each subdomain introduced in [9] is based on the extracellular potential computed from a suitable extension of the Monodomain model. More precisely, we let $\mathbf{D}_\varepsilon = \mathbf{D}_e - \frac{\sigma_e^l}{\sigma_i^l}\mathbf{D}_i$. The model estimator is computed at the subdomain level as

$$\zeta_j^2 = \int_{\Omega_j} \nabla u_M \frac{\sigma_i^l \mathbf{D}_\varepsilon}{\sigma_i^l+\sigma_e^l}\left(\mathbf{D}_i^{-1} + \mathbf{D}_e^{-1}\right)\frac{\sigma_i^l \mathbf{D}_\varepsilon}{\sigma_i^l+\sigma_e^l}\nabla u_M \, dx. \tag{11}$$

The value ζ_j^2 is an upper bound for the error in Ω_j between the two models in a $H^1(\Omega_j)$-type seminorm depending on $\mathbf{D}_i$ and $\mathbf{D}_e$. The Bidomain model is then activated in Ω_j whenever ζ_j^2 exceeds a given threshold τ_j, depending on the size of the subdomain. Computing ζ_j^2 requires one matrix-vector and one scalar product,

and we denote by $\mathbf{K}_\varepsilon$ the stiffness matrix associated with (11). More details on this estimator, that we do not report for the sake of space, can be found in [9].

2.3 The model adaptive algorithm

Preprocessing

(i) Split the computational domain into non-overlapping subregions Ω_j ($j = 1,..,N$).
(ii) Identify the interfaces Γ_{ij} between subdomains Ω_i and Ω_j.
(iii) Assemble the local matrices $\mathbf{B}^j_{uu}$, $\mathbf{B}^j_{ue}$, $\mathbf{B}^j_{eu}$, $\mathbf{B}^j_{ee}$, and $\mathbf{K}^j_\varepsilon$.
(iv) Assemble the interface mass matrices $\mathbf{M}_{\Gamma_{ij}}$.
(v) Compute the incomplete ILU factorization of the local $\mathbf{B}^j_{uu}$ and $\mathbf{B}^j_{ee}$ matrices.

Runtime (time step $t^n \to t^{n+1}$)

(i) Run a Monodomain simulation at time t^{n+1} over the whole domain Ω.
(ii) Evaluate the model estimator and compute the local indicator $\zeta_j^2 = (u_M^j)^T \mathbf{K}^j_\varepsilon u_M^j$.
(iii) For all Ω_j ($j = 1,..,N$) such that $\zeta_j^2 > \tau_j$, activate Bidomain.
(iv) Run the Optimized Schwarz Algorithm using the solution computed in Step 1 as initial guess. A few iterations are usually enough.
(v) Advance to time t^{n+1}.

3 Preliminary numerical results

Numerical results in this section have the purpose to show the effectiveness of the model adaptive method: for this reason we consider here only 2D simulations. The numerical tests are run in Matlab® 7.5. The Bidomain problems are solved by a flexible GMRES (f-GMRES) right preconditioned by the Block-triangular preconditioner introduced in [4], while the Monodomain problems are solved by a CG preconditioned by an ILU factorization.
We consider the strip $\Omega = [0,3] \times [0,1]$ subdivided into the three noverlapping subdomains $\Omega_i = [i-1, i] \times [0,1]$, $i = 1,2,3$. The fibers are oriented with the principal direction perpendicular to the interfaces, and we impose a stimulus on the whole left boundary of Ω_1. The well known Rogers-McCulloch ionic model [13] is used.

We plot in Figure 2 the wavefront position at different times (top row), and the activated subdomains (bottom row) during depolarization. The advantage of the model adaptive approach resides in solving only cheap Monodomain problems for the large majority of time steps (in a genuine Monodomain setting, without extensions that were needed by the Hybrodomain approach). In Table 1 we report the relative CPU gain over a whole heartbeat duration (450ms) for the model adaptive strategy with respect to the Optimized Schwarz algorithm introduced in [5].

	u_{OSB}	u_{MA}
CPU Time	1.000	0.37

Table 1 Relative CPU time: u_{OSB} and u_{MA} computed with 2 Schwarz iterations.

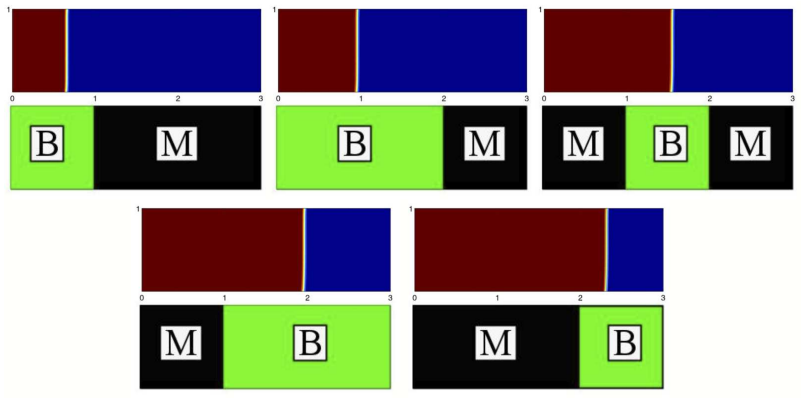


Fig. 2 Propagation of the membrane potential (in red the excited region, top row), and the activated Bidomain region (in green and marked by "B", bottom row).

A more detailed presentation of the method will be the subject of a forthcoming work. Further work needs to be done to identify the proper trade-off between the number of subdomains, and the size of the Bidomain region surrounding the wave-front, and to properly handle the processors load balance in a parallel architecture. Also, dynamical allocation of tasks is under investigation to properly balance, in real problems, the load of each processor in the parallel solver.

References

1. R. H. Clayton, O. M. Bernus, E. M. Cherry, H. Dierckx, F. H. Fenton, L. Mirabella, A. V. Panfilov, F. B. Sachse, G. Seemann, H. Zhang, Models of cardiac tissue electrophysiology: Progress, challenges and open questions, *Progress in biophysics and molecular biology* 104, pp. 22-48, 2011.
2. P. Colli Franzone, L. Pavarino, G. Savaré. Computational electrocardiology: mathematical and numerical modeling, in *Complex Systems in Biomedicine* - A. Quarteroni, L. Formaggia, and A. Veneziani, editors. Springer, Milan, 2006.
3. M.J. Gander. Optimized Schwarz methods. *SIAM J. Num. Anal.*, 44(2), pp. 699–731, 2006.
4. L. Gerardo-Giorda, L. Mirabella, F. Nobile, M. Perego, and A. Veneziani. A model-based block-triangular preconditioner for the Bidomain system in electrocardiology. *J. Comp. Phys.*, 228, pp. 3625–3639, 2009.
5. L. Gerardo-Giorda and M. Perego, Optimized Schwarz Methods for the Bidomain system in electrocardiology *M2AN*, Vol. 47 (2), pp 583–608, 2013.
6. L. Gerardo-Giorda, M. Perego, and A. Veneziani. Optimized Schwarz coupling of Bidomain and Monodomain models in electrocardiology. *M2AN*, Vol. 45 (2), pp. 309-334, 2011.

7. J. Le Grice, B.H. Smaill, and P.J. Hunter. Laminar structure of the heart: a mathematical model. *Am. J. Physiol.*, 272 (Heart Circ. Physiol.)(41):H2466–H2476, 1995.
8. G.T. Lines, M.L. Buist, P. Grottum, A.J. Pullan, J. Sundnes, and A. Tveito. Mathematical models and numerical methods for the forward problem in cardiac electrophysiology *Comput. Visual. Sci*,5:215-239, 2003.
9. L. Mirabella, F. Nobile, and A. Veneziani. An a posteriori error estimator for model adaptivity in electrocardiology. *Comp. Meth. Appl. Mech. Engrg.*, Vol 200 (37-40), pp. 2727–2737, 2011.
10. L. F. Pavarino and S. Scacchi. Multilevel additive Schwarz preconditioners for the Bidomain reaction-diffusion system. *SIAM J. Sci. Comp.*, 31(1):420–443, 2008.
11. M. Pennacchio and V. Simoncini. Algebraic multigrid preconditioners for the bidomain reactiondiffusion system. *Appl. Num. Math.*, 59(12):3033–3050, 2009.
12. M. Potse, B. Dubé, J. Richer, and A. Vinet. A comparison of Monodomain and Bidomain Reaction-Diffusion models for Action Potential Propagation in the Human Heart. *IEEE Trans. Biomed. Eng.*, 53(12):2425–2435, 2006.
13. J. Rogers and A. McCulloch. A collocation-Galerkin finite element model of cardiac action potential propagation. *IEEE Transactions on Biomedical Engineering*, 41:743–757, 1994.
14. F. B. Sachse. *Computational Cardiology*. Springer, Berlin, 2004.
15. E.J. Vigmond, R. Weber dos Santos, A.J. Prassl, M. Deo, and G. Plank. Solvers for the caridac bidomain equations. *Progress in Biophysics and Molecular Biology*, 96(1-3):3–18, 2008.
16. N. Trayanova. Defibrillation of the heart: insights into mechanisms from modelling studies. *Experimental Physiology*, 91: 323–337, 2006.

A new interface cement equilibrated mortar method with Ventcel conditions

Caroline Japhet[1], Yvon Maday[2], and Frédéric Nataf[3]

1 Introduction

For many applications in mechanics or fluid dynamics, one need to use different discretizations in different regions of the computational domain to match with the physical scales. Mortar methods [2] are domain decomposition techniques based on a weak coupling between subdomains and enable the use of nonconforming grids. On the other hand, optimized Schwarz methods [4, 11, 9, 7, 5], based on Robin or Ventcel transmission conditions and motivated by the physics of the underlying problem, greatly enhance the information exchange between subdomains and lead to robust and fast algorithms. Moreover, the Ventcel conditions reduce dramatically the convergence factor of the Schwarz algorithm compared to Robin conditions [7, 5].

In the finite element case, the NICEM method [6, 8], a new interface cement using Robin conditions and corresponding to an equilibrated mortar approach (i.e. there is no master and slave sides) has been developed for Schwarz type methods.

In this paper we extend this approach to Ventcel conditions.

We first consider the problem at the continuous level: find u such that

$$(Id-\Delta)u = f \quad \text{in } \Omega \tag{1}$$
$$u = 0 \quad \text{on } \partial\Omega, \tag{2}$$

where Ω is a $\mathscr{C}^{1,1}$ (or convex polygon in 2D or polyhedron in 3D) domain of $\mathbb{R}^d$, $d = 2$ or 3, and f is given in $L^2(\Omega)$. We assume that Ω is decomposed into K non-overlapping subdomains: $\overline{\Omega} = \cup_{k=1}^{K} \overline{\Omega}^k$. We suppose that the subdomains Ω^k, $1 \le k \le K$ are either $\mathscr{C}^{1,1}$ or polygons in 2D or polyhedrons in 3D. Let $\boldsymbol{n}_k$ be the outward normal from Ω^k. We also assume that this decomposition is geometrically conforming. We introduce $\Gamma^{k,\ell}$ the interface of two adjacent subdomains, $\Gamma^{k,\ell} = \partial\Omega^k \cap \partial\Omega^\ell$. An optimized Schwarz algorithm for problem (1)-(2) is

$$\begin{aligned} (Id-\Delta)u_k^{n+1} &= f && \text{in } \Omega^k \\ u_k^{n+1} &= 0 && \text{on } \partial\Omega^k \cap \partial\Omega \\ \mathscr{B}_{k,\ell}(u_k^{n+1}) &= \mathscr{B}_{k,\ell}(u_\ell^n) && \text{on } \Gamma^{k,\ell} \end{aligned}$$

[1] Université Paris 13, LAGA, UMR 7539, F-93430, Villetaneuse, France. INRIA Paris-Rocquencourt, BP 105, 78153 Le Chesnay, France, e-mail: japhet@math.univ-paris13.fr .[2] UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France. Institut Universitaire de France. Brown Univ, Division of Applied Maths, Providence, RI, USA, e-mail: maday@ann.jussieu.fr .[3] UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France, e-mail: nataf@ann.jussieu.fr

where $(\mathscr{B}_{k,\ell})_{1\leq k,\ell\leq K,k\neq\ell}$ is the chosen transmission operator on the interface between subdomains Ω^k and Ω^ℓ:

$$\textbf{Robin case: } \mathscr{B}_{k,\ell}\varphi = \partial_{\boldsymbol{n}}\varphi + \alpha\varphi$$
$$\textbf{Ventcel case: } \mathscr{B}_{k,\ell}\varphi = \partial_{\boldsymbol{n}}\varphi + \alpha\varphi - \beta\Delta_{\tau_{k,\ell}}\varphi,$$

where $\Delta_{\tau_{k,\ell}}$ stands for the Laplace-Beltrami operator on $\Gamma^{k,\ell}$, and $\alpha,\beta>0$ are given. In order to match Ventcel conditions in the non-conforming discrete case, we need to introduce a new independent entity representing the normal derivative of the solution on the interface as in the NICEM method [6, 8]. We thus use a Petrov Galerkin approach instead of Galerkin approximations as in standard mortar methods.

In Sect. 2 we recall the method at the continuous level. Then in Sect. 3, we present the method in the non-conforming discrete case and the discrete algorithm with Ventcel transmission conditions. We finally present in Sect. 4 simulations for two and twenty-five subdomains. The numerical analysis will be done in future paper.

2 Definition of the problem

The variational statement of the problem (1)-(2) is: Find $u\in H_0^1(\Omega)$ such that

$$\int_\Omega (\nabla u\nabla v + uv)\,dx = \int_\Omega fv dx, \quad \forall v\in H_0^1(\Omega). \tag{3}$$

We introduce the space $H_*^1(\Omega^k)$ defined by

$$H_*^1(\Omega^k) = \{\varphi\in H^1(\Omega^k),\ \varphi = 0 \text{ over } \partial\Omega\cap\partial\Omega^k\}.$$

In order to glue non-conforming grids with Ventcel transmission conditions, denoting by $\underline{v}$ the K-tuple $(v_1,...,v_K)$, we introduce the following constrained space,

$$\mathscr{V} = \{(\underline{v},\underline{q})\in \left(\prod_{k=1}^K H_*^1(\Omega^k)\right)\times\left(\prod_{k=1}^K H^{-1/2}(\partial\Omega^k)\right),$$
$$v_k = v_\ell \text{ and } q_k = -q_\ell \text{ over } \Gamma^{k,\ell},\ \forall k,\ell\}. \tag{4}$$

Then, problem (3) is equivalent to the following one [8]: Find $(\underline{u},\underline{p})\in\mathscr{V}$ such that

$$\sum_{k=1}^K\int_{\Omega^k}(\nabla u_k\nabla v_k + u_kv_k)\,dx - \sum_{k=1}^K {}_{H^{-1/2}(\partial\Omega^k)}\langle p_k, v_k\rangle_{H^{1/2}(\partial\Omega^k)}$$
$$= \sum_{k=1}^K\int_{\Omega^k} f_kv_k dx, \quad \forall\underline{v}\in\prod_{k=1}^K H_*^1(\Omega^k).$$

Being equivalent with (1)-(2), where $p_k = \partial_{\boldsymbol{n}_k}u$ over $\partial\Omega^k$, this problem is well posed.

Let us describe the method in the non-conforming discrete case.

3 Non-conforming discrete case with Ventcel conditions

3.1 Local problem

We introduce now the discrete spaces. Each Ω^k is provided with its own mesh $\mathscr{T}_h^k$, such that $\overline{\Omega}^k = \cup_{T\in\mathscr{T}_h^k} T,\ 1 \le k \le K$. For $T \in \mathscr{T}_h^k$, let h_T be the diameter of T and h the discretization parameter: $h = \max_{1\le k\le K} h_k$ with $h_k = \max_{T\in\mathscr{T}_h^k} h_T$. We suppose that $\mathscr{T}_h^k$ is uniformly regular and that the sets belonging to the meshes are of simplicial type (triangles or tetrahedra). Let $\mathscr{P}_M(T)$ denote the space of all polynomials defined over T of total degree less than or equal to M. The finite elements are of lagrangian type, of class $\mathscr{C}^0$. We define over each Ω^k two conforming spaces Y_h^k and X_h^k by : $Y_h^k = \{v_{h,k} \in \mathscr{C}^0(\overline{\Omega}^k),\ v_{h,k|T} \in \mathscr{P}_M(T),\ \forall T \in \mathscr{T}_h^k\}$, $X_h^k = \{v_{h,k} \in Y_h^k,\ v_{h,k|\partial\Omega^k\cap\partial\Omega} = 0\}$. The space of traces over each $\Gamma^{k,\ell}$ of elements of Y_h^k is a finite element space denoted by $\mathscr{Y}_h^{k,\ell}$. With each interface $\Gamma^{k,\ell}$, we associate a subspace $\tilde{W}_h^{k,\ell}$ of $\mathscr{Y}_h^{k,\ell}$ in the same spirit as in the mortar element method [2] in 2D or [3, 1] for a P_1-discretization in 3D.

More precisely, let $\mathscr{T}$ be the restriction to $\Gamma^{k,\ell}$ of the triangulation $\mathscr{T}_h^k$. In 2D, $\mathscr{T}$ has two end points that we denote as $x_0^{k,\ell}$ and $x_n^{k,\ell}$ that belong to the set of vertices of the corresponding triangulation of $\Gamma^{k,\ell}$: $x_0^{k,\ell}, x_1^{k,\ell}, ..., x_{n-1}^{k,\ell}, x_n^{k,\ell}$. The space $\tilde{W}_h^{k,\ell}$ is then the subspace of those elements of $\mathscr{Y}_h^{k,\ell}$ that are polynomials of degree $\le M-1$ over both $[x_0^{k,\ell}, x_1^{k,\ell}]$ and $[x_{n-1}^{k,\ell}, x_n^{k,\ell}]$.

In 3D, we suppose that all the vertices of the boundary of $\Gamma^{k,\ell}$ are connected to zero, one, or two vertices in the interior of $\Gamma^{k,\ell}$. Let $\mathscr{V}$, $\mathscr{V}_0$, $\partial\mathscr{V}$ denote respectively the set of all the vertices of $\mathscr{T}$, the vertices in the interior of $\Gamma^{k,\ell}$, and the vertices on the boundary of $\Gamma^{k,\ell}$. Let $S(\mathscr{T})$ be the space of piecewise linear functions with respect to $\mathscr{T}$ which are continuous on $\Gamma^{k,\ell}$ and vanish on its boundary. We denote by $\Phi_a,\ a \in \mathscr{V}$ the finite element basis functions. Thus, $S(\mathscr{T}) =$ span $\{\Phi_a : a \in \mathscr{V}_0\}$. For $a \in \mathscr{V}$, let $\sigma_a := \bigcup\{T \in \mathscr{T} : a \in T\}$, $\mathscr{N}_a := \{b \in \mathscr{V}_0 : b \in \sigma_a\}$, and $\mathscr{N} := \cup_{a\in\partial\mathscr{V}} \mathscr{N}_a$. Let $\mathscr{T}_c$ be the set of triangles $T \in \mathscr{T}$ which have all their vertices on the boundary of $\Gamma^{k,\ell}$. For $T \in \mathscr{T}_c$, we denote by c_T the only vertex of T that has no interior neighbor. Let $\mathscr{N}_c$ denote the vertices a_T of $\mathscr{N}$ which belong to a triangle adjacent to a triangle $T \in \mathscr{T}_c$. We introduce $\hat{\Phi}_a$ defined as follows:

$$\hat{\Phi}_a := \begin{cases} \Phi_a, & a \in \mathscr{V}_0 \setminus \mathscr{N} \\ \Phi_a + \displaystyle\sum_{b\in\partial\mathscr{V}\cap\sigma_a} A_{b,a}\Phi_b, & a \in \mathscr{N} \setminus \mathscr{N}_c \\ \Phi_{a_T} + \displaystyle\sum_{b\in\partial\mathscr{V}\cap\sigma_{a_T}} A_{b,a_T}\Phi_b + \Phi_{c_T}, & a = a_T \in \mathscr{N}_c \end{cases}.$$

The weights are defined such that [3]: $A_{c,a} + A_{c,b} = 1$ and $|T_{2,b}|A_{c,a} = |T_{2,a}|A_{c,b}$, for all boundary nodes $c \in \partial\mathscr{V}$ connected to two interior nodes a and b. Here $T_{2,a}$ (resp. $T_{2,b}$) denote the adjacent triangle to abc having a (resp. b) as a vertex and its

two others vertices on $\partial\mathscr{V}$. For all boundary nodes $c \in \partial\mathscr{V}$ connected to only one interior node a, the weights are $A_{c,a} = 1$.

The space $\tilde{W}_h^{k,\ell}$ is then defined by $\tilde{W}_h^{k,\ell} :=$ span $\{\hat{\Phi}_a,\ a \in \mathscr{V}_0\}$. Then $\tilde{W}_h^k$ is the product space of the $\tilde{W}_h^{k,\ell}$ over each ℓ such that $\Gamma^{k,\ell} \neq \emptyset$.

We introduce now the discrete problem. Let $\nabla_{\tau_{k,\ell}}$ be the gradient operator on $\Gamma^{k,\ell}$. We define the discrete constrained space as follows:

$$\mathscr{V}_h = \{(\underline{u}_h, \underline{p}_h) \in \left(\prod_{k=1}^K X_h^k\right) \times \left(\prod_{k=1}^K \tilde{W}_h^k\right),$$

$$\int_{\Gamma^{k,\ell}} ((p_{h,k} + \alpha u_{h,k}) - (-p_{h,\ell} + \alpha u_{h,\ell}))\psi_{h,k,\ell} + \int_{\Gamma^{k,\ell}} \beta \nabla_{\tau_{k,\ell}}(u_{h,k} - u_{h,\ell}) \nabla_{\tau_{k,\ell}} \psi_{h,k,\ell}$$
$$- \int_{\partial\Gamma_{k,\ell}} \beta \left(\nabla_{\tau_{k,\ell}} u_{h,k} - \nabla_{\tau_{k,\ell}} u_{h,\ell}\right) \psi_{h,k,\ell} = 0,\ \forall \psi_{h,k,\ell} \in \tilde{W}_h^{k,\ell}\}, \quad (5)$$

and the discrete problem is the following one : Find $(\underline{u}_h, \underline{p}_h) \in \mathscr{V}_h$ such that

$$\forall \underline{v}_h = (v_{h,1}, \ldots v_{h,K}) \in \prod_{k=1}^K X_h^k,$$

$$\sum_{k=1}^K \int_{\Omega^k} \left(\nabla u_{h,k} \nabla v_{h,k} + u_{h,k} v_{h,k}\right) dx - \sum_{k=1}^K \int_{\partial\Omega^k} p_{h,k} v_{h,k} ds = \sum_{k=1}^K \int_{\Omega^k} f_k v_{h,k} dx. \quad (6)$$

Let us describe the algorithm in the discrete case.

3.2 Iterative algorithm

We restrict ourselves to the presentation of the algorithm in 2D.

The recommended approach to find the solution of the previous discrete problem is a GMRES acceleration [12] of the iterative Schwarz algorithm. For the sake of clarity, let us present the plain Jacobi algorithm applied to the discrete Schwarz algorithm : let $(u_{h,k}^n, p_{h,k}^n) \in X_h^k \times \tilde{W}_h^k$ be a discrete approximation of (u,p) in Ω^k at step n. Then, $(u_{h,k}^{n+1}, p_{h,k}^{n+1})$ is the solution in $X_h^k \times \tilde{W}_h^k$ of

$$\int_{\Omega^k} \left(\nabla u_{h,k}^{n+1} \nabla v_{h,k} + u_{h,k}^{n+1} v_{h,k}\right) dx - \int_{\partial\Omega^k} p_{h,k}^{n+1} v_{h,k} ds = \int_{\Omega^k} f_k v_{h,k} dx,\ \forall v_{h,k} \in X_h^k, \quad (7)$$

$$\int_{\Gamma^{k,\ell}} \left((p_{h,k}^{n+1} + \alpha u_{h,k}^{n+1}) \psi_{h,k,\ell} + \beta \nabla_{\tau_{k,\ell}} u_k^{n+1} \nabla_{\tau_{k,\ell}} \psi_{h,k,\ell}\right) - \int_{\partial\Gamma_{k,\ell}} \beta \nabla_{\tau_{k,\ell}} u_{h,k}^{n+1} \psi_{h,k,\ell}$$
$$= \int_{\Gamma^{k,\ell}} \left((-p_{h,\ell}^n + \alpha u_{h,\ell}^n) \psi_{h,k,\ell} + \beta \nabla_{\tau_{k,\ell}} u_\ell^n \nabla_{\tau_{k,\ell}} \psi_{h,k,\ell}\right)$$
$$- \int_{\partial\Gamma_{k,\ell}} \beta \nabla_{\tau_{k,\ell}} u_{h,\ell}^n \psi_{h,k,\ell}, \quad \forall \psi_{h,k,\ell} \in \tilde{W}_h^{k,\ell}. \quad (8)$$

An initial guess $(g_{k,\ell})$ is given on each interface $\Gamma_{k,\ell}$, and by convention for the first iterate, the right-hand side in (8) is given by $g_{k,\ell}$.

4 Numerical results

In this part, we consider a P_1 finite element approximation. Problem (6) is a square linear system, invertible in the various numerical tests we performed, the results presented below being some of them. We study the numerical error analysis for problem (6), as well as the convergence of the algorithm (7)-(8) with Ventcel compared to Robin (i.e. $\beta = 0$) transmissions conditions.

We consider the initial problem with exact solution $u(x,y) = x^3y^2 + sin(xy)$. The domain is the unit square $\Omega = (0,1) \times (0,1)$.

We decompose Ω into non-overlapping subdomains with meshes generated in an independent manner. On Fig. 1, we consider the case of 2 non-conforming meshes (on the left), and the case of 25 non-conforming meshes (on the right). In the sequel, for the error curves versus h, the computed solution is the solution at convergence of the discrete algorithm (7)-(8), with a stopping criterion on the L^2 norm of the jumps of the interface conditions that must be smaller than 10^{-14}.

4.1 Choice of the Ventcel parameters α,β

In our numerical results, the Ventcel parameters are obtained by minimizing the convergence factor (depending on the mesh size in that case). In the conforming two subdomains case, with constant mesh size h and an interface of length L, the optimal theoretical values of the Ventcel parameters α,β which minimize the convergence

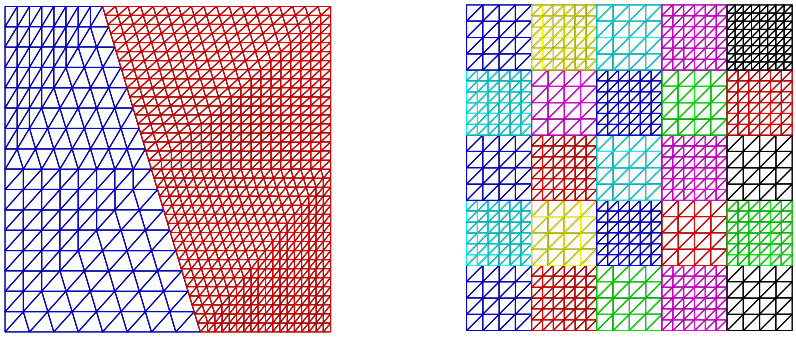


Fig. 1 Nonconforming domain decomposition in 2 domains (left), and 25 domains (right)

factor at the continuous level are [5]:

$$\alpha^* = \frac{k_{max}^2\sqrt{k_{min}^2+1}-k_{min}^2\sqrt{k_{max}^2+1}}{\sqrt{2(k_{max}^2-k_{min}^2)}\left(\left(\sqrt{k_{max}^2+1}-\sqrt{k_{min}^2+1}\right)\left((k_{max}^2+1)\sqrt{k_{min}^2+1}-(k_{min}^2+1)\sqrt{k_{max}^2+1}\right)\right)^{\frac{1}{4}}}$$
$$\beta^* = \frac{\sqrt{k_{max}^2+1}-\sqrt{k_{min}^2+1})^{\frac{3}{4}}}{\sqrt{2(k_{max}^2-k_{min}^2)}\left((k_{max}^2+1)\sqrt{k_{min}^2+1}-(k_{min}^2+1)\sqrt{k_{max}^2+1}\right)^{\frac{1}{4}}}, \tag{9}$$

where k_{min} and k_{max} are respectively the minimum and maximum frequencies which can be represented on a grid with mesh size h, given by $k_{min} = \frac{1}{L}$ and $k_{max} = \frac{\pi}{h}$. In the non-conforming case, the mesh size is different for each side of the interface. Thus, we consider the parameters given by (9) with $h = h_m$ denoted by (α^m, β^m), or with $h = h_M$ denoted by (α^M, β^M), where h_m and h_M are respectively the smallest and highest step size on the interface. We consider also the Robin case with the optimal theoretical value given by [5]: $\alpha_R^* = \left((\frac{\pi}{L})^2+1\right)\left((\frac{\pi}{h_M})^2+1\right)^{\frac{1}{4}}$.

4.2 Two subdomains case

In this part we consider the 2 non-conforming meshes on the left of Fig. 1. As the problem (6) depends on α,β, we consider two cases: $(\alpha,\beta) = (\alpha_m,\beta_m)$ (case (m)) and $(\alpha,\beta) = (\alpha_M,\beta_M)$ (case (M)). In order to observe the error versus h, a computed solution (solution of (6)) corresponds to the solution at convergence of (7)-(8). The solution with $(\alpha,\beta) = (\alpha_m,\beta_m)$ is different from the one with $(\alpha,\beta) = (\alpha_M,\beta_M)$. We represent on Fig. 2 (left), for both cases, the relative H^1 error (defined as in [8]), and the relative L^2 error versus the mesh size h, in logarithmic scale. We start from the 2 non-conforming meshes and then refine successively each mesh by dividing the mesh size by two. We observe similar results for both cases. The results show that the relative H^1 error tends to zero at the same rate as the mesh size h. We also observe that the relative L^2 error tends to zero at the same rate as h^2. We represent on Fig. 2 (right) the asymptotic performance with optimized Ventcel (i.e. $\alpha = \alpha_M$, $\beta = \beta_M$) or Robin (i.e. $\alpha = \alpha_R^*$, $\beta = 0$) conditions, for the Schwarz algorithm (7)-(8) and for the GMRES algorithm. We simulate directly the error equations, $f = 0$, and use a random initial guess so that all the frequency components are present. We plot the number n_* of iterations (taken to reduce the error by a factor 10^{-6}) versus h on a log-log plot. The numerical results show the asymptotic behavior predicted by the analysis given in [5]:

- $n_* = O(h^{\frac{1}{2}})$ for Robin (i.e. $\alpha = \alpha_R^*$, $\beta = 0$) with Schwarz as an iterative solver,
- $n_* = O(h^{\frac{1}{4}})$ for Robin with GMRES (i.e. Schwarz used as a preconditioner),
- $n_* = O(h^{\frac{1}{4}})$ for Ventcel (i.e. $\alpha = \alpha_M$, $\beta = \beta_M$) with Schwarz as an iterative solver,
- $n_* = O(h^{\frac{1}{8}})$ for Ventcel with GMRES.

We also observe that using Krylov acceleration (GMRES) improves the asymptotic performance by a square root.

4.3 Twenty-five subdomains case

We now consider the 25 non-conforming meshes on the right of Fig. 1.

In order to observe the H^1 error, each computed solution corresponds to the solution at convergence of (7)-(8). We represent on Fig. 3 (left) the relative H^1 error versus the mesh size h in logarithmic scale. We start from the 25 non-conforming meshes and then refine successively each mesh by dividing the mesh size by two. The results show that the relative H^1 error tends to zero at the same rate as the mesh size h. On Fig. 3 (right), we study the performance of the algorithm (7)-(8) with Ventcel and Robin transmission conditions. We simulate directly the error equations, $f = 0$, and use a random initial guess on the interfaces. We plot the H^1 and L^∞ errors versus the number of iterations. We observe that the number of iterations

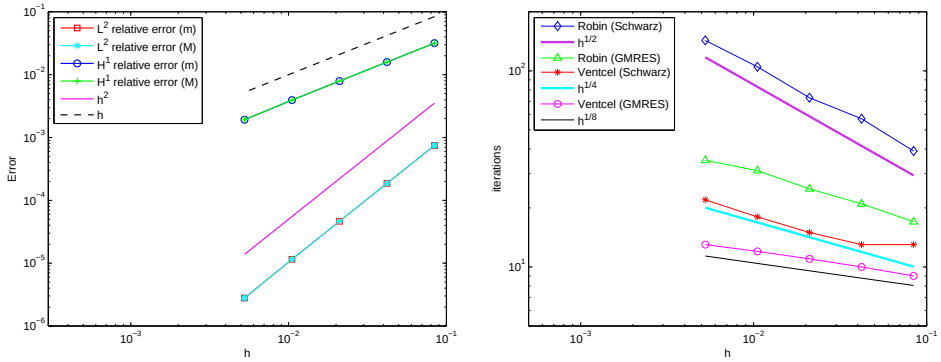


Fig. 2 Decomposition in 2 subdomains: error analysis versus h (left), and asymptotic number of iterations required by the method with optimized Robin or Ventcel conditions, when the method is used as iterative solver, or used as preconditioner for a Krylov method (GMRES)

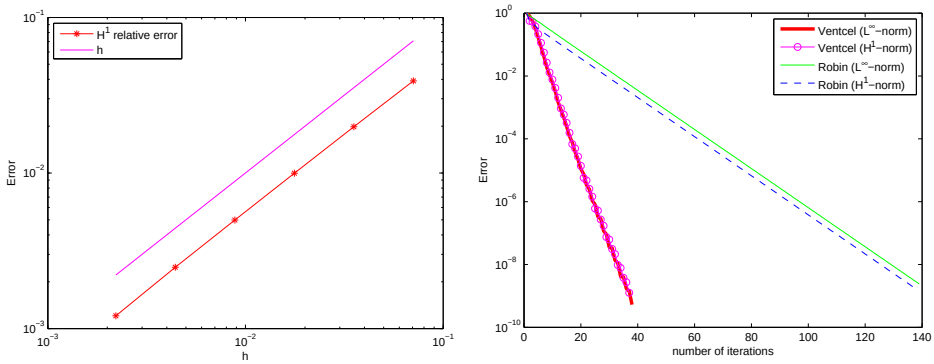


Fig. 3 Decomposition in 25 subdomains: H^1 error versus h (left), and error versus iterations (in the H^1 and L^∞ norms) with optimized Robin or Ventcel conditions

to obtain an error smaller than 10^{-6} is by a factor 4 higher with optimized Robin conditions compared to optimized Ventcel conditions. The results are similar for the H^1 and L^∞ errors.

References

1. Ben Belgacem, F., Maday, Y.: Coupling spectral and finite elements for second order elliptic three-dimensional equations. SIAM J. Numer. Anal. **36 (4)**, 1234–1263 (1999)
2. Bernardi, C., Maday, Y., Patera, A.: A new nonconforming approach to domain decomposition: the mortar element method. Brezis, H. and Lions, J.L., Pitman (1989)
3. Braess, D., Dahmen, W.: Stability estimates of the mortar finite element method for 3-dimensional problems. East-West J. Numer. Math. **6 (4)**, 249–263 (1998)
4. Després, B.: Domain decomposition method and the Helmholtz problem. Mathematical and Numerical aspects of wave propagation phenomena, SIAM, 44–52 (1991)
5. Gander, M.J.: Optimized Schwarz Methods. SIAM J. Numer. Anal. **44 (2)**, 699–731 (2006)
6. Gander, M.J., Japhet, C., Maday, Y, Nataf, F.: A New Cement to Glue Nonconforming Grids with Robin Interface Conditions : The Finite Element Case. In R. Kornhuber, R.H.W. Hoppe, J. Périaux, O. Pironneau, O.B. Widlund, J. Xu (eds) Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering **40**, 259–266. Springer (2004)
7. Japhet, C.: Optimized Krylov-Ventcell Method. Application to Convection-Diffusion Problems. In P. Bjorstad, M. Espedal, D. Keyes (eds) Decomposition Methods in Sciences and Engineering, International Conference on Domain Decomposition Methods, 3-8 june 1996, Bergen (Norway), 382–389. Springer (1998)
8. Japhet, C., Maday, Y, Nataf, F.: A New Interface Cement Equilibrated Mortar (NICEM) method with Robin interface conditions: the P_1 finite element case. M^3AS (2013)
9. Nataf, F., Rogier, F: Factorization of the Convection-Diffusion Operator and the Schwarz Algorithm. M^3AS **1** 67–93 (1995)
10. Nataf, F., Rogier, F., de Sturler, E.: Domain Decomposition Methods for Fluid Dynamics. Navier-Stokes Equations and Related Nonlinear Analysis, Sequeira, A. (eds), 367–376. Plenum Press Corporation (1995)
11. Gastaldi, F., Gastaldi, L., Quarteroni, A.: Adaptative Domain Decomposition Methods for Advection dominated Equations. East-West J. Numer. Math. **4**, 165–206 (1996)
12. Saad, Y.: Iterative Methods for Sparse Linear Systems. SIAM (2003)

FETI-DP methods for Optimal Control Problems

Roland Herzog[1] and Oliver Rheinbach[2]

1 Introduction

We consider FETI-DP domain decomposition methods for optimal control problems of the form

$$\min_{y,u} \frac{1}{2}\int_\Omega (y(x)-y_d(x))^2\,dx + \frac{\alpha}{2}\int_\Omega (u(x))^2\,dx, \tag{1}$$

where $y \in V$ denotes the unknown state and $u \in U$ the unknown control, subject to a PDE constraint

$$a(y,v) = (f,v)_0 + (u,v)_0 \qquad \text{for all } v \in V. \tag{2}$$

The function y_d denotes a given desired state and $\alpha > 0$ a cost parameters. By $(\cdot,\cdot)_0$, we denote the standard L_2 inner product. In this paper, $a(\cdot,\cdot)$ will be the bilinear form associated with linear elasticity, i.e.,

$$a(y,v) = (2\mu\varepsilon(y),\varepsilon(v))_0 + (\lambda \operatorname{div} y, \operatorname{div} v)_0, \tag{3}$$

where μ, and λ are the Lamé parameters.

The state (displacement field) is sought in $V = H^1_0(\Omega,\partial\Omega_D)^2 = \{y \in H^1(\Omega)^2 : y = 0 \text{ on } \partial\Omega_D\}$, where $\Omega \subset \mathbf{R}^2$ and $\partial\Omega_D$ is part of its boundary. For simplicity, we consider the case of volume control, i.e., $U = L_2(\Omega)^2$.

Dual-primal FETI methods were first introduced by Farhat, Lesoinne, Le Tallec, Pierson, and Rixen [3] and have successfully scaled to 10^5 processor cores [6]. In [8] a first convergence bound for scalar problems in 2D was provided. Numerical scalability for FETI-DP methods applied to linear elasticity problems was first proven in [7].

Balancing Neumann-Neumann domain decomposition methods for the optimal control of scalar problems have been considered in Heinkenschloss and Nguyen [5, 4]. There, local optimal control problems on non-overlapping subdomains are considered and a Balancing Neumann-Neumann preconditioner is constructed for the indefinite Schur complement. Multigrid methods have, of course, also been considered for optimal control problems, see, e.g., [10]. A review of block approaches to optimal control problems can be found in [9]. A recent block approach can be found in [11].

[1] Fakultät für Mathematik, Technische Universität Chemnitz, 09107 Chemnitz, Germany, e-mail: roland.herzog@mathematik.tu-chemnitz.de .[2] Mathematisches Institut, Universität zu Köln, Weyertal 86-90, 50931 Köln, Germany, e-mail: orheinba@mi.uni-koeln.de

We discretize y by $P1$ finite elements, u by $P0$ finite elements and obtain the discrete problem

$$\min_{y,u} \frac{1}{2} y^T M y + \frac{\alpha}{2} u^T Q u - c^T y \tag{4}$$

$$\text{s.t.} \qquad Ay = f + Nu. \tag{5}$$

2 Discrete Problem and Domain Decomposition

The necessary and sufficient optimality conditions are given by the discrete system

$$\begin{bmatrix} M & 0 & A^T \\ 0 & \alpha Q & -N^T \\ A & -N & 0 \end{bmatrix} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} c \\ 0 \\ f \end{bmatrix} \tag{6}$$

where $A \in \mathbf{R}^{n\times n}$, $Q \in \mathbf{R}^{m\times m}$, $M \in \mathbf{R}^{n\times n}$. Here, $A = A^T = (a(\varphi_i, \varphi_j))_{i,j}$ is a stiffness matrix, whereas $Q = (\langle \psi_i, \psi_j \rangle)_{i,j}$, $M = (\langle \psi_i, \psi_j \rangle)_{i,j}$ and $N = (\langle \varphi_i, \psi_j \rangle)_{i,j}$ are mass matrices. We will denote the block system (6) by

$$Kx = b. \tag{7}$$

We decompose Ω into N nonoverlapping subdomains $\Omega_i, i = 1, \ldots, N$, i.e. $\overline{\Omega} = \bigcup_{i=1}^N \overline{\Omega}_i$, $\Omega_i \cap \Omega_j = \emptyset$ if $i \neq j$. Each subdomain is the union of shape-regular finite element cells with matching nodes across the interface, $\Gamma := \bigcup_{i\neq j} \partial\Omega_i \cap \partial\Omega_j$, where $\partial\Omega_i, \partial\Omega_j$ are the boundaries of Ω_i, Ω_j, respectively.

For each subdomain, we assemble the local problem $K^{(i)}$, which represents the discrete optimality system for (1)–(2), restricted to the subdomain Ω_i. Let us denote, for each subdomain, the variables that are on the subdomain interface by an index Γ and the interior unknowns by I. Note that the interior variables also comprise the variables on the Neumann boundary $\partial\Omega \setminus \partial\Omega_D$. In block form, we can now write the subdomain problem matrices $K^{(i)}, i = 1, \ldots, N$ as

$$K^{(i)} = \begin{bmatrix} M^{(i)} & 0 & A^{(i)T} \\ 0 & \alpha Q^{(i)} & -N^{(i)T} \\ A^{(i)} & -N^{(i)} & 0 \end{bmatrix} = \left[\begin{array}{cc|c|cc} M_{II}^{(i)} & M_{I\Gamma}^{(i)} & 0 & A_{II}^{(i)} & A_{I\Gamma}^{(i)} \\ M_{I\Gamma}^{(i)T} & M_{\Gamma\Gamma}^{(i)} & 0 & A_{I\Gamma}^{(i)T} & A_{\Gamma\Gamma}^{(i)} \\ \hline 0 & 0 & \alpha Q_{II}^{(i)} & -N_{II}^{(i)T} & -N_{\Gamma I}^{(i)T} \\ \hline A_{II}^{(i)} & A_{I\Gamma}^{(i)} & -N_{II}^{(i)} & 0 & 0 \\ A_{I\Gamma}^{(i)T} & A_{\Gamma\Gamma}^{(i)} & -N_{\Gamma I}^{(i)} & 0 & 0 \end{array}\right]. \tag{8}$$

We define the block matrices

$$K_{II}^{(i)} = \begin{bmatrix} M_{II}^{(i)} & 0 & A_{II}^{(i)} \\ 0 & \alpha Q_{II}^{(i)} & -N_{II}^{(i)} \\ A_{II}^{(i)} & -N_{II}^{(i)} & 0 \end{bmatrix}, \; K_{\Gamma\Gamma}^{(i)} = \begin{bmatrix} M_{\Gamma\Gamma}^{(i)} & A_{\Gamma\Gamma}^{(i)} \\ A_{\Gamma\Gamma}^{(i)} & 0 \end{bmatrix}, \; K_{I\Gamma}^{(i)} = \begin{bmatrix} M_{I\Gamma}^{(i)} & A_{I\Gamma}^{(i)} \\ 0 & -N_{\Gamma I}^{(i)T} \\ A_{I\Gamma}^{(i)} & 0 \end{bmatrix}. \tag{9}$$

Following the approach of FETI-type methods a continuity constraint $Bx = 0$ is introduced to enforce the continuity of y and p across each interface Γ. The introduction of Lagrange multipliers λ then leads to the FETI master system

$$\begin{bmatrix} K^{(1)} & & & \widehat{B}^{(1)} \\ & \ddots & & \vdots \\ & & K^{(N)} & \widehat{B}^{(N)} \\ \widehat{B}^{(1)} & \dots & \widehat{B}^{(N)} & 0 \end{bmatrix} \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(N)} \\ \lambda \end{bmatrix} = \begin{bmatrix} b^{(1)} \\ \vdots \\ b^{(N)} \\ 0 \end{bmatrix}. \tag{10}$$

In the context of our optimal control problem, $\widehat{B}^{(i)}$ is of the form $\widehat{B}^{(i)} = \left[B_y^{(i)} \middle| 0 \middle| B_p^{(i)} \right]$. Note that it is not appropriate to enfore continuity for the control variable u, since it is an algebraic variable and has been discretized by discontinuous elements.

In dual-primal FETI methods the continuity constraint is enforced on a subset of the variables on the interface Γ by partial finite element assembly. These variables are denoted by the index Π (primal). Here, for our 2D problems, we use primal vertex variables. For the remaining interface variables, the continuity is enforced by Lagrange multipliers. Such interface variables are denoted by the index Δ (dual). We thus write the matrices $M^{(i)}, A^{(i)}, N^{(i)}$ appearing in (8) in the form

$$M^{(i)} = \begin{bmatrix} M_{II}^{(i)} & M_{I\Delta}^{(i)} & M_{I\Pi}^{(i)} \\ M_{I\Delta}^{(i)T} & M_{\Delta\Delta}^{(i)} & M_{\Delta\Pi}^{(i)} \\ M_{I\Pi}^{(i)T} & M_{I\Delta}^{(i)T} & M_{\Pi\Pi}^{(i)} \end{bmatrix}, A^{(i)} = \begin{bmatrix} A_{II}^{(i)} & A_{I\Delta}^{(i)} & A_{I\Pi}^{(i)} \\ A_{I\Delta}^{(i)T} & A_{\Delta\Delta}^{(i)} & A_{\Delta\Pi}^{(i)} \\ A_{I\Pi}^{(i)T} & A_{I\Delta}^{(i)T} & A_{\Pi\Pi}^{(i)} \end{bmatrix}, N^{(i)} = \begin{bmatrix} N_{II}^{(i)} \\ N_{\Delta I}^{(i)} \\ N_{\Pi I}^{(i)} \end{bmatrix}, \tag{11}$$

and $Q^{(i)} = Q_{II}^{(i)}$. Inserting this block form into (8), we obtain the block form of $K_{\Pi\Pi}^{(i)}$,

$$K_{\Pi\Pi}^{(i)} = \begin{bmatrix} M_{\Pi\Pi}^{(i)} & A_{\Pi\Pi}^{(i)T} \\ A_{\Pi\Pi}^{(i)} & 0 \end{bmatrix}. \tag{12}$$

For the assembly of the primal variables y_Π and p_Π, we define the combined assembly operator $\widehat{R}_\Pi^{(i)T}$, i.e., we obtain for the assembled global matrix $\widetilde{K}_{\Pi\Pi}$

$$\begin{aligned} \widetilde{K}_{\Pi\Pi} = \widehat{R}_\Pi^T K_{\Pi\Pi} \widehat{R}_\Pi &= \left[\widehat{R}_\Pi^{(1)T}, \dots, \widehat{R}_\Pi^{(N)T} \right] \begin{bmatrix} K_{\Pi\Pi}^{(1)} & & 0 \\ & \ddots & \\ 0 & & K_{\Pi\Pi}^{(N)} \end{bmatrix} \begin{bmatrix} \widehat{R}_\Pi^{(1)} \\ \\ \widehat{R}_\Pi^{(N)} \end{bmatrix} \\ &= \sum_{i=1}^N \widehat{R}_\Pi^{(i)T} K_{\Pi\Pi}^{(i)} \widehat{R}_\Pi^{(i)} = \sum_{i=1}^N \begin{bmatrix} R_\Pi^{(i)T} & 0 \\ 0 & R_\Pi^{(i)T} \end{bmatrix} \begin{bmatrix} M_{\Pi\Pi}^{(i)} & A_{\Pi\Pi}^{(i)T} \\ A_{\Pi\Pi}^{(i)} & 0 \end{bmatrix} \begin{bmatrix} R_\Pi^{(i)} & 0 \\ 0 & R_\Pi^{(i)} \end{bmatrix} \\ &= \begin{bmatrix} \widetilde{M}_{\Pi\Pi} & \widetilde{A}_{\Pi\Pi}^T \\ \widetilde{A}_{\Pi\Pi} & 0 \end{bmatrix}. \end{aligned} \tag{13}$$

The partially assembled system matrix is then

$$\widetilde{K} = \begin{bmatrix} K_{BB}^{(1)} & & & \widetilde{K}_{B\Pi}^{(1)} \\ & \ddots & & \vdots \\ & & K_{BB}^{(N)} & \widetilde{K}_{B\Pi}^{(N)} \\ \widetilde{K}_{B\Pi}^{(1)T} & \dots & \widetilde{K}_{B\Pi}^{(N)T} & \widetilde{K}_{\Pi\Pi} \end{bmatrix} \tag{14}$$

with the blocks

$$K_{BB}^{(i)} = \begin{bmatrix} M_{II}^{(i)} & M_{I\Delta}^{(i)} & 0 & A_{II}^{(i)} & A_{I\Delta}^{(i)} \\ M_{I\Delta}^{(i)T} & M_{\Delta\Delta}^{(i)} & 0 & A_{I\Delta}^{(i)T} & A_{\Delta\Delta}^{(i)} \\ 0 & 0 & \alpha Q_{II}^{(i)} & -N_{II}^{(i)T} & -N_{\Delta I}^{(i)T} \\ A_{II}^{(i)} & A_{I\Delta}^{(i)} & -N_{II}^{(i)} & 0 & 0 \\ A_{I\Delta}^{(i)T} & A_{\Delta\Delta}^{(i)} & -N_{\Delta I}^{(i)} & 0 & 0 \end{bmatrix}, \tag{15}$$

and

$$\begin{aligned} \widetilde{K}_{B\Pi}^{(i)T} &= \begin{bmatrix} \widetilde{M}_{I\Pi}^{(i)T} & \widetilde{M}_{\Delta\Pi}^{(i)T} & 0 & \widetilde{A}_{I\Pi}^{(i)T} & \widetilde{A}_{\Delta\Pi}^{(i)T} \\ \widetilde{A}_{I\Pi}^{(i)T} & \widetilde{A}_{\Delta\Pi}^{(i)T} & \widetilde{N}_{I\Pi}^{(i)T} & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} R_{\Pi}^{(i)T} & 0 \\ 0 & R_{\Pi}^{(i)T} \end{bmatrix} \begin{bmatrix} \widetilde{M}_{I\Pi}^{(i)T} & \widetilde{M}_{\Delta\Pi}^{(i)T} & 0 & \widetilde{A}_{I\Pi}^{(i)T} & \widetilde{A}_{\Delta\Pi}^{(i)T} \\ \widetilde{A}_{I\Pi}^{(i)T} & \widetilde{A}_{\Delta\Pi}^{(i)T} & \widetilde{N}_{I\Pi}^{(i)T} & 0 & 0 \end{bmatrix}. \end{aligned} \tag{16}$$

Now, we can formulate the FETI-DP master system,

$$\begin{bmatrix} \widetilde{K} & \widehat{B}^T \\ \widehat{B} & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{b} \\ 0 \end{bmatrix}, \qquad u \in \mathbf{R}^n, \lambda \in \mathbf{R}^m, \tag{17}$$

from which the solution of the original finite element problem (6) can be obtained by averaging the solution $\tilde{x}$ from (17) in the interface variables. Here, the jump operator $\widehat{B}$ only acts on the variables y_Δ and p_Δ. The vectors $\tilde{x}$ and $\tilde{b}$ have the form

$$\begin{aligned} x^T &= \left[[y_I^{(i)T}, y_\Delta^{(i)T}, u_I^{(i)T}, p_I^{(i)T}, p_\Delta^{(i)T}], \dots, [y_I^{(N)T}, y_\Delta^{(N)T}, u_I^{(N)T}, p_I^{(N)T}, p_\Delta^{(N)T}], [\tilde{y}_\Pi^T, \tilde{p}_\Pi^T] \right] \\ b^T &= \left[[c_I^{(i)T}, c_\Delta^{(i)T}, 0, f_I^{(i)T}, f_\Delta^{(i)T}], \dots, [c_I^{(N)T}, c_\Delta^{(N)T}, 0, f_I^{(N)T}, f_\Delta^{(N)T}], [\tilde{c}_\Pi^T, \tilde{f}_\Pi^T] \right] \end{aligned}$$

After the elimination of x in (17) it remains to solve a system

$$F\lambda = d \tag{18}$$

where F is symmetric indefinite, i.e., with positive and negative eigenvalues, by a suitable Krylov subspace method. The FETI-DP coarse problem is

$$\widetilde{S}_{\Pi\Pi} = \widetilde{K}_{\Pi\Pi} - \sum_{i=1}^{N} \widetilde{K}_{B\Pi}^{(i)} \widetilde{K}_{\Pi\Pi}^{(i)} \widetilde{K}_{B\Pi}^{(i)T}. \tag{19}$$

To define the Dirichlet preconditioner, we consider the block submatrices of $K^{(i)}$ defined in (9),

$$K^{(i)} = \begin{bmatrix} K_{II}^{(i)} & K_{\Gamma I}^{(i)T} \\ K_{\Gamma I}^{(i)} & K_{\Gamma\Gamma}^{(i)} \end{bmatrix}. \tag{20}$$

Let us define the Schur complement

$$S_{\Gamma\Gamma} = \sum_{i=1}^{N} (K_{\Gamma\Gamma}^{(i)} - K_{\Gamma I}^{(i)} (K_{II}^{(i)})^{-1} K_{\Gamma I}^{(i)T}) = \sum_{i=1}^{N} S_{\Gamma\Gamma}^{(i)}, \tag{21}$$

which can be computed completely in parallel. The Dirichlet preconditioner is then given in matrix form by

$$M^{-1} = B_D \widehat{R}_\Gamma^T S_{\Gamma\Gamma} \widehat{R}_\Gamma B_D^T = \sum_{i=1}^{N} B_D^{(i)} \widehat{R}_\Gamma^{(i)T} S_{\Gamma\Gamma}^{(i)} \widehat{R}_\Gamma^{(i)} B_D^{(i)T}, \tag{22}$$

where B_D is a variant of the jump operator B scaled by the inverse multiplicity of the node. The matrices $R_\Gamma^{(i)}$ are simple restriction operators which restrict the nonprimal degrees of freedom of a subdomain to the interface, i.e. $\widehat{R}_\Gamma^{(i)} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}$, if the variables are numbered $[I, \Delta]$ on the right hand side and $[\Delta, \Pi]$ on the left hand side of the operator.

3 Well-posedness of the local problems

In [4] the well-posedness of the local subdomain problems for the balancing Neumann-Neumann method was considered. These considerations are also valid for FETI-1-type methods. In contrast to FETI-1 and Balancing Neumann-Neumann methods the coarse problems of the more recent FETI-DP and BDDC methods are constructed from partial finite element assembly.

We therefore briefly comment on the well-posedness of the subdomain problems, i.e. the local blocks $K_{BB}^{(i)}$ in (14), as well as the coarse problem (19). Each block $K_{BB}^{(i)}$ represents a discrete optimality system local to the subdomain Ω_i. In contrast to the original problem (2), natural (stress) boundary conditions are imposed on $\partial\Omega_i$ for the state y, except in the (few) primal degrees of freedom on the interface boundary, and except for the degrees of freedom on $\partial\Omega_i \cap \partial\Omega_D$, where Dirichlet conditions apply. These conditions are sufficient to exclude rigid body motions. Consequently, the local elasticity system (the four A blocks in (15) combined), is well posed, and thus it is straightforward to show that also the optimality system is well posed, whence $K_{BB}^{(i)}$ non-singular. The non-singularity of the total matrix $\widetilde{K}$ in (14) can be shown along the same lines. And thus the non-singularity of the Schur complement (19) follows.

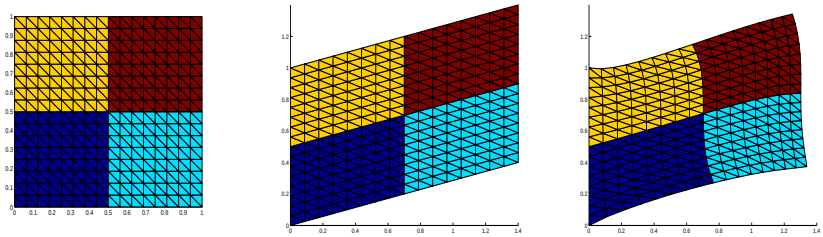


Fig. 1 Model problem: Undeformed configuration, desired state, and solution computed using FETI-DP.

Finally, (21) is well defined since $K_{II}^{(i)}$ is non-singular. Note that each $K_{II}^{(i)}$ represents a discrete optimality system with all-Dirichlet boundary conditions on $\partial\Omega_i$ for the state and adjoint states, with these boundary degrees of freedom removed.

4 Numerical Results

Here we will report on the use of GMRES applied to the symmetric indefinite FETI-DP system (18), using the symmetric indefinite Dirichlet preconditioner (22). Note that there is no theory for the convergence of GMRES in this situation. The numerical results are nevertheless very encouraging. We also report on the convergence of QMR. The stopping criterion is the relative reduction of the preconditioned residual by 10 orders of magnitude. In [5, 4] a symmetric QMR was used for the Neumann-Neumann method. The numerical results are nevertheless very encouraging. The iteration counts using QMR and GMRES are very similar.

We consider the volume control of a linear elastic problem on the unit square. The desired displacement y_d is a obtained from applying a linear transformation to the unit square, i.e., $y_d(x,y) = (\frac{2}{5}x, \frac{2}{5}y)^T$; see Fig. 1. The Dirichlet boundary is on the left. The material data is $E = 1$ (Young's modulus) and $\nu = 0.3$ (Poisson's ratio) in all cases, which are related to the Lamé constants via $E = \frac{\mu(2\mu+3\lambda)}{\mu+\lambda}$ and $\nu = \frac{\lambda}{2(\mu+\lambda)}$.

We numerically observe scalability with respect to the number of subdomains as known for CG in the symmetric positive case, i.e., the number of iterations approaches a limit for an increasing number of subdomains N if H/h is maintained fixed, see Tab. 1. Moreover the number of iterations grows only weakly with H/h for a fixed number of subdomains N, see Tab. 2. In Tab. 3 we see that the methods shows robustness with respect to α. In Tab. 4 we report on the strong parallel scalability of the largest problem from Tab. 2 using the GMRES implementation from PETSc [1]. We have used UMFPACK 4.3 [2] for the solution of the subdomain problems.

DIRICHLET PRECONDITIONER - Weak Scaling - GMRES and QMR

N	#Points	#Elem	#gmres	#qmr	#Points	#Elem	#gmres	#qmr	#Points	#Elem	#gmres	#qmr
	$H/h=2$				$H/h=4$				$H/h=8$			
2×2	25	32	**8**	**9**	81	128	**11**	**11**	289	512	**13**	**14**
4×4	81	128	**14**	**14**	289	512	**19**	**20**	1089	2048	**25**	**27**
6×6	169	288	**15**	**16**	625	1152	**22**	**24**	2401	4608	**30**	**32**
8×8	289	512	**15**	**16**	1089	2048	**24**	**25**	4225	8192	**32**	**34**
10×10	441	800	**16**	**16**	1681	3200	**24**	**25**	6561	12800	**33**	**36**
12×12	625	1152	**16**	**17**	2401	4608	**25**	**26**	9409	18432	**34**	**38**
16×16	1089	2048	**16**	**17**	4225	8192	**25**	**26**	16641	32768	**35**	**38**
20×20	1681	3200	**16**	**17**	6561	12800	**25**	**26**	25921	51200	**36**	**39**
24×24	2401	4608	**16**	**18**	9409	18432	**25**	**26**	37249	73728	**36**	**39**
28×28	3249	6272	**16**	**18**	12769	25088	**26**	**26**	50625	100352	**36**	**40**
32×32	4225	8192	**16**	**18**	16641	32768	**26**	**27**	66049	131072	**37**	**40**
36×36	5329	10368	**16**	**18**	21025	41472	**26**	**27**	83521	165888	**37**	**41**
40×40	6561	12800	**16**	**18**	25921	51200	**26**	**27**	103041	204800	**37**	**41**
48×48	9409	18432	**16**	**18**	37249	73728	**26**	**27**	148225	294912	**37**	**41**
56×56	12769	25088	**16**	**18**	50625	100352	**26**	**27**	201601	401408	**37**	**41**
64×64	16641	32768	**16**	**19**	66049	131072	**26**	**27**	263169	524288	**37**	**41**

Table 1 Weak scaling. The number of GMRES and QMR iterations is scalable with respect to the number of subdomains, i.e., it is bounded independently of N. $\alpha = 0.01$. Material parameters $E = 1$, $\nu = 0.3$. The iteration is stopped when the preconditioned residual has been reduced by 10 orders of magnitudes. The largest problem has $2\,101\,252 = 4\times 263\,169 + 2\times 524\,288$ d.o.f.

References

1. Balay, S., Brown, J., , Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.3, Argonne National Laboratory (2012)
2. Davis, T.A.: A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. ACM Trans. Math. Software **30**(2), 167–195 (2004). DOI 10.1145/992200.992205
3. Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., Rixen, D.: FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. Internat. J. Numer. Methods Engrg. **50**(7), 1523–1544 (2001). DOI 10.1002/nme.76
4. Heinkenschloss, M., Nguyen, H.: Balancing Neumann-Neumann methods for elliptic optimal control problems. In: Domain decomposition methods in science and engineering, *Lect. Notes Comput. Sci. Eng.*, vol. 40, pp. 589–596. Springer, Berlin (2005)
5. Heinkenschloss, M., Nguyen, H.: Neumann-Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems. SIAM J. Sci. Comput. **28**(3), 1001–1028 (2006). DOI 10.1137/040612774
6. Klawonn, A., Rheinbach, O.: Highly scalable parallel domain decomposition methods with an application to biomechanics. ZAMM Z. Angew. Math. Mech. **90**(1), 5–32 (2010). DOI 10.1002/zamm.200900329. URL `http://dx.doi.org/10.1002/zamm.200900329`
7. Klawonn, A., Widlund, O.B.: Dual-primal FETI methods for linear elasticity. Comm. Pure Appl. Math. **59**(11), 1523–1572 (2006). DOI 10.1002/cpa.20156. URL `http://dx.doi.org/10.1002/cpa.20156`
8. Mandel, J., Tezaur, R.: On the convergence of a dual-primal substructuring method. Numer. Math. **88**(3), 543–558 (2001). DOI 10.1007/s211-001-8014-1
9. Mathew, T.P., Sarkis, M., Schaerer, C.E.: Analysis of block matrix preconditioners for elliptic optimal control problems. Numer. Linear Algebra Appl. **14**(4), 257–279 (2007)

DIRICHLET PRECONDITIONER - GMRES and QMR

H/h	#Points	#Elem	#gmres	#qmr	#Points	#Elem	#gmres	#qmr	#Points	#Elem	#gmres	#qmr
	$N = 2\times 2$				$N = 3\times 3$				$N = 4\times 4$			
2	25	32	**8**	**9**	49	72	**12**	**13**	81	128	**14**	**14**
4	81	128	**11**	**11**	169	288	**16**	**17**	289	512	**19**	**20**
6	169	288	**12**	**12**	361	648	**18**	**19**	625	1152	**23**	**24**
8	289	512	**13**	**14**	625	1152	**20**	**21**	1089	2048	**25**	**27**
12	625	1152	**14**	**14**	1369	2592	**23**	**25**	2401	4608	**28**	**31**
16	1089	2048	**14**	**15**	2401	4609	**25**	**27**	4225	8192	**31**	**34**
24	2401	4608	**16**	**16**	5329	10368	**28**	**30**	9409	18432	**35**	**36**
32	4225	8192	**16**	**17**	9409	18432	**29**	**30**	16641	32768	**37**	**38**
48	9409	18432	**17**	**18**	21025	41472	**32**	**33**	37249	73728	**40**	**43**
64	16641	32768	**18**	**19**	37249	73728	**33**	**35**	66049	131072	**43**	**45**
96	37249	73728	**19**	**19**	83521	165888	**34**	**38**	148225	294912	**46**	**50**
128	66049	131072	**19**	**20**	148225	294912	**36**	**39**	263169	524288	**49**	**52**

Table 2 The number of GMRES and QMR iterations grows only weakly with the subdomain size. $\alpha = 0.01$. Material parameters $E = 1$, $\nu = 0.3$. The iteration is stopped when the preconditioned residual has been reduced by 10 orders of magnitudes. The largest problem has $2\,102\,452 = 2\times 2\times 263\,169 + 524\,288$ d.o.f.

DIRICHLET PRECONDITIONER - GMRES and QMR

N	H/h	α	#gmres	#qmr
8×8	4	1	**19**	**20**
8×8	4	0.1	**22**	**22**
8×8	4	0.01	**24**	**25**
8×8	4	0.001	**23**	**24**
8×8	4	0.0001	**19**	**21**

Table 3 Dependence on α. The preconditioner is robust with respect to the choice of the cost parameter $\alpha > 0$.

#Cores	N	H/h	#Points	#Elem	d.o.f.	#gmres	Time
1	4×4	64	66049	131072	526340	49	89.7*s*
2	4×4	64	66049	131072	526340	49	45.6*s*
4	4×4	64	66049	131072	526340	49	23.9*s*
8	4×4	64	66049	131072	526340	49	14.2*s*
16	4×4	64	66049	131072	526340	49	10.7*s*

Table 4 Strong parallel scalability on a 16 core Opteron 8380 server (2.5 Ghz) for one of the problems from Tab. 2.

10. Schöberl, J., Simon, R., Zulehner, W.: A robust multigrid method for elliptic optimal control problems. SIAM J. Numer. Anal. **49**(4), 1482–1503 (2011)
11. Schöberl, J., Zulehner, W.: Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems. SIAM J. Matrix Anal. Appl. **29**(3), 752–773 (electronic) (2007)

Domain decomposition methods in Feel++

Abdoulaye Samaké[1], Vincent Chabannes[1], Christophe Picard[1], and Christophe Prud'homme[2]

1 Introduction

Libraries to solve problems arising from partial differential equations (PDEs) through generalized Galerkin methods are a common tool among mathematicians and engineers. However, most libraries end up specializing in a type of equation, e.g. Navier-Stokes or linear elasticity models, or a specific type of numerical method, e.g. finite elements. The increasing complexity of differential models and the implementation of state of the art robust numerical methods, demand from scientific computing platforms general and clear enough languages to express such problems and provide a wealth of solution algorithms available in a minimal amount of code but maximum mathematical control. There are many freely available libraries which offer the capabilities described previously to a certain extent. To name a few: the Freefem software family [6, 9], the Fenics project [10], Getdp [8] or Getfem++ [17], or libraries or frameworks such as deal.II (C++) [2], Sundance (C++) [11], Analysa (Scheme) [1].

The library we present in this paper, called FEEL++, *Finite Element Embedded Language in* C++, see [14, 15], provides also a clear and easy to use interface to solve complex PDE systems. It aims at bringing the scientific community a tool for the implementation of advanced numerical methods and high performance computing. Some recent applications of FEEL++ to multiphysics problems can be found in the literature, see *e.g.* [13, 7, 5].

FEEL++ relies on a so-called *domain specific embedded language* (DSEL) designed to closely match the Galerkin mathematical framework. In computer science, DS(E)Ls are used to partition complexity and in our case the DSEL splits low level mathematics and computer science on one side leaving the FEEL++ developer to enhance them and high level mathematics as well as physical applications to the other side which are left to the FEEL++ user. This enables using FEEL++ for teaching purposes, solving complex problems with multiple physics and scales or rapid prototyping of new methods, schemes or algorithms.

The DSEL on FEEL++ provides access to powerful, yet with a simple and seamless interface, tools such as interpolation or the clear translation of a wide range of variational formulations into the variational embedded language. Combined with this robust engine, lie also state of the art arbitrary order finite elements — including

[1]Laboratoire Jean Kuntzmann, Université Joseph Fourier Grenoble 1, BP53 38041 Grenoble Cedex 9, France, Tel.: +33476635497, Fax: +33476631263, e-mail: {abdoulaye.samake}{vincent.chabannes}{christophe.picard}@imag.fr .[2] Université de Strasbourg / CNRS, IRMA / UMR 7501, Strasbourg, F-67000, France, e-mail: prudhomme@unistra.fr

handling high order geometrical approximations, — high order quadrature formulas and robust nodal configuration sets. The tools at the user's disposal grant the flexibility to implement numerical methods that cover a large combination of choices from meshes, function spaces or quadrature points using the same integrated language and control at each stage of the solution process the numerical approximations.

This paper presents our ongoing work on building a computational framework for domain decomposition methods in FEEL++ including overlapping and nonoverlapping Schwarz methods (conforming and non-conforming) and mortar method. The complete examples are available in FEEL++ sources. Note that examples using the three fields method are also available in FEEL++.

The framework main objectives consist in (*i)* reproducing and comparing easily several of methods in the literature (*ii)* developing a teaching and research programming environment (*iii)* providing the methods at the functional level or at the algebraic level. In this context we have also developed also two alternatives: one which lets the user control the MPI communications and one which hides completely the MPI communications.

2 Schwarz Methods

Let Ω be a domain of $\mathbb{R}^d$, $d = 1,2,3$, and $\partial\Omega$ its boundary. We look for u the solution of the problem:

$$Lu = f \quad \text{in} \quad \Omega, \quad u = g \quad \text{on} \quad \partial\Omega \tag{1}$$

where L is a linear partial differential operator, and f and g are given functions. Let $\Omega_i (i = 1,...,N,\ N \in \mathbb{N},\ N \geq 2)$ the subdomain partitions of Ω such that $\overline{\Omega} = \cup_{i=1}^{N} \overline{\Omega}_i$ and $\Gamma_{ij} = \partial\Omega_i \cap \overline{\Omega}_j$ the interface between neighboring subdomains Ω_i and Ω_j. We denote $\mathscr{V}_{\Omega_i}$ the set of neighbors subdomains of Ω_i. In the case of nonoverlapping subdomains $\Gamma_{ij} = \Gamma_{ji}$. We are interested in the overlapping and nonoverlapping alternating Schwarz methods[16, 19] as solver in the general nonmatching grids and arbitrary number of subdomains. The generic Schwarz additive algorithm is given by (2) where u_i^0 is known on Γ_{ij}, $j \in \mathscr{V}_{\Omega_i}$, $k \geq 1$ the Schwarz iteration index and C_i is a partial differential operator.

$$Lu_i^k = f \text{ in} \quad \Omega_i, \quad u_i^k = g \text{ on} \quad \partial\Omega_i \setminus \Gamma_{ij}, \quad C_i u_i^k = C_i u_j^{k-1} \text{on} \quad \Gamma_{ij} \tag{2}$$

The algorithm (2) extends easily to the multiplicative version of Schwarz methods and treats different types of artificial boundary conditions such as Dirichlet-Dirichlet (DD), Dirichlet-Neumann (DN), Neumann-Neumann (NN) and Robin-Robin (RR) (see [20, 16, 19]) according the choice of the operator C_i that is assumed linear in our case. The above algorithm can also adapt to relaxation techniques(see [16]) necessary for the convergence of some types of interface conditions such as DN and NN without overlap.

In the following subsections 2.1 and 2.2, we discuss two different approaches for Schwarz methods in FEEL++ namely with explicit communications and with seamless communications. In the first approach, we deal different types of Schwarz methods(Additive, Multiplicative, with(out) Relaxation) with different artificial boundary conditions(DD, DN, NN, RR) while having the ability to process (non-)conforming meshes as well as being able to control the size of the overlap between neighboring subdomains. In the second approach, we use the parallel data structures of FEEL++ and the algebraic domain decomposition framework provided by PETSC.

2.1 Explicit Communication Approach

The Schwarz methods are used as solvers and the communications are handled explicitly by the user. Implementation-wise we use PETSC sequentially even though the code is parallel using `mpi` communicators. It requires explicitly sending and receiving complex data structures such as mesh data structures and elements of functions space(traces). A sequential interpolation operator is also used to make the transfer between the grids (overlapping or not, conforming or not). In this case each subdomain creates locally its mesh and its function space, the matrices and vectors associated to the discretization process are completely local.

The variational formulation of the problem (2) in the simplest form ($L := -\Delta$) in the subdomain Ω_i at iteration number k using Nitsche's method (see [12]) in the case of weak Dirichlet-Dirichlet artificial boundary conditions ($C_i = C_j = Id,\ j \in \mathscr{V}_{\Omega_i}$) is given by: find $u_i^k \in H^1(\Omega_i)$ such that $a(u_i^k, v) = l(v)\ \forall v \in H^1(\Omega_i)$ where

$$a(u_i^k, v) := \int_{\Omega_i} \nabla u_i^k \cdot \nabla v + \int_{\partial\Omega_i} -\frac{\partial u_i^k}{\partial n} v - \frac{\partial v}{\partial n} u_i^k + \frac{\gamma}{h} u_i^k v \tag{3}$$

$$l(v) := \int_{\Omega_i} fv + \int_{\partial\Omega_i \setminus \Gamma_{ij}} \Big(-\frac{\partial v}{\partial n} + \frac{\gamma}{h} v \Big) g + \sum_{j \in \mathscr{V}_{\Omega_i}} \int_{\Gamma_{ij}} \Big(-\frac{\partial v}{\partial n} + \frac{\gamma}{h} v \Big) u_j^{k-1} \tag{4}$$

where γ is a penalization parameter and h the maximum mesh size.

Other variants of artificial boundary conditions such as Dirichlet-Neumann($C_i = Id,\ C_j = \partial/\partial n,\ j \in \mathscr{V}_{\Omega_i}$), Neumann-Neumann ($C_i = C_j = \partial/\partial n,\ j \in \mathscr{V}_{\Omega_i}$) and Robin-Robin($C_i = C_j = (\partial/\partial n) + Id,\ j \in \mathscr{V}_{\Omega_i}$) are also treated. In the above variational formulation, only the terms colored in red in (4) requires communications between neighboring subdomains for each Schwarz iteration and interpolation between the grids. Note that the assembly of the other terms of the variational formulation is done once and is purely local. We make use of `Boost.MPI` and `Boost.Serialization` to ease the transfer of FEEL++ complex data structures such as meshes and (elements of) function spaces.

Listing 1 Feel++ snippet code for parallel Schwarz algorithm

```
// Create local mesh and function space on subdomain number i
```

```
auto mesh = createGMSHMesh(_mesh=mesh_type, ...);
auto Xh = space_type::New(mesh);
std::vector<mpi::request> reqs; // vector of Boost.MPI requests
for(int j=0, j< Nneighbors, ++j){
  // Extract trace mesh on interface number j
  trace_mesh_send[j]=mesh->trace(markedfaces(mesh,j));
  // Exchange trace mesh with neighbor subdomain number j
  auto req1=comm.isend( j,i,trace_mesh_send[j] );
  auto req2=comm.irecv( j,j,trace_mesh_recv[j] );
  reqs.push_back(req1); reqs.push_back(req2);
 } mpi::wait_all(reqs.begin(), reqs.end());// wait all requests
for(int j=0, j< Nneighbors, ++j){
  // Create trace function space for interface number j
  TXh[j] = trace_space_type::New(trace_mesh_recv[j]);
  // Create interpolation operator from Xh to TXh[j]
  opI[j]=operatorInterpolation(Xh,TXh[j]); }
while(!convergence) { // Schwarz iterations
  reqs.clear();
  for(int j=0, j< Nneighbors, ++j){
    // Non conforming interpolation for interface number j
    opI[j]->apply(solution,trace_solution_send[j]);
    // Exchange trace solution with neighbor subdomain number j
    auto req1=comm.isend( j,i,trace_solution_send[j] );
    auto req2=comm.irecv( j,j,trace_solution_recv[j] );
    reqs.push_back(req1); reqs.push_back(req2);
  } mpi::wait_all(reqs.begin(), reqs.end());// wait all requests
    // Update right hand side for each schwarz iteration
  for(int j=0, j< Nneighbors, ++j){
    form1( _test=Xh,_vector=F ) +=
      integrate(elements(trace_mesh_send[j]),
        -grad(v)*N()*idv(trace_solution_recv[j])
        +penaldir*idv(trace_solution_recv[j])*id(v)/hFace()); }
  solve(); }
```

To illustrate our implementation of the Schwarz method, we consider the problem (1) over a partition over the domain $\Omega = [0,1]^2$ into 128 overlapping subdomains (16×8) with non matching meshes. The boundary condition and the source write $g(x,y) = 0$ and $f(x,y) = \exp(-10xy)\cos(\frac{3\pi}{8})\sin(xy)$.

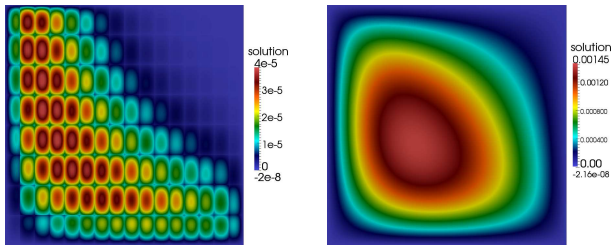


Fig. 1 Numerical solutions obtained by Schwarz parallel additive algorithm in 2D on 128 processors(1 subdomain/processor): First schwarz iteration(Left) and solution at convergence(Right)

The numerical solutions in Figure 1 are obtained using $\mathbb{P}_2$ Lagrange elements. The precision of the numerical solver is fixed to $1e-7$. The mesh size is 0.01 in each subdomain and the size of the overlap is 0.02 but we don't ensure that the grids are conforming. The total number of degree of freedom is 153600. The number of Schwarz iterations to convergence is 130 and the relative L^2 error $\|u-u_h\| = 1.164901e-06$. The listing 1 illustrates some aspects of the Schwarz algorithm using the `Feel++` language.

2.2 Seamless Communication Approach

Here we consider the domain decomposition methods with seamless communications in FEEL++. We provide a parallel data framework: we start with automatic mesh partitioning using GMSH(Chaco/Metis) — adding information about ghosts cells with communication between neighbor partitition; — then FEEL++ data structures are parallel such as meshes, (elements of) function spaces — create a parallel degrees of freedom table with local and global views; — and finally we use the PETSC Krylov subspace solvers(KSP) coupled with PETSC preconditioners such as Block-Jacobi, ASM, GASM. The last preconditioner is an additive variant of the Schwarz alternating method for the case of many subregion, see [19]. For each subpreconditioners(in the subdomains), PETSC allows to choose in the wide range of sequential preconditioners such, ilu, jacobi, ml.

To illustrate this, we perform a strong scalability test with a Laplace problem in 3D using P3 Lagrange elements (about 8 Millions degrees of freedom). The listing 2 corresponds to the code that allowed us to realize this test. The speedup displayed in table 1 corresponds to the assembly plus the solve times. We can see that the scaling is good except for the last configuration where the local problems is too small.

Listing 2 Laplacian Solver using continuous approximation spaces and PETSc in parallel

```
/* Create parallel function space and some associated elements */
auto Xh = space_type::New( _mesh=mesh );
/* Create the parallel matrix and vector of linear system */
auto A = backend()->newMatrix(_test=Xh, _trial=Xh);
auto F = backend()->newVector(Xh);
/* Parallel assembly of the right hand side */
form1( _test=Xh, _vector=F )=
    integrate( _range=elements( mesh ), _expr=f*id( v ) )
/* Parallel assembly of the global matrix */
form2( _test=Xh, _trial=Xh, _matrix=A ) =
    integrate( _range=elements( mesh ),
               _expr=gradt(u)*trans(grad(v)) );
/* Apply Dirichlet boundary conditions strongly */
form2( _test=Xh, _trial=Xh, _matrix=A ) +=
  on( _range=boundaryfaces(mesh),
      _element=u,_rhs=F, _expr=g );
/* solve system using PETSc parallel solvers/preconditioners */
backend()->solve( _matrix=A, _solution=u, _rhs=F );
```

Table 1 Strong scalability test

Number of Cores	Absolute Times	Speedup
1024	41.2	1
2048	18.2	2.26
4096	10	4.12
8192	7	5.88

3 Mortar Method

Consider the problem (1) where $L := -\Delta$ and homogeneous Dirichlet boundary conditions. We assume that Ω is partitioned into two nonoverlapping subdomains and it is a d-dimensional domain ($d = 2,3$), with a Lipschitz boundary $\partial\Omega$. We also assume that f belongs to $L^2(\Omega)$. The main idea of this method is to enforce the weak continuity between the solutions on each subdomain. This is achieved by introducing a Lagrange multiplier corresponding to this connection constraint [3].

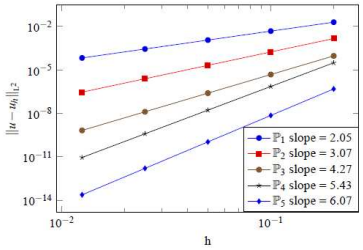


Fig. 2 Convergence results for Mortar Element Method in 2D with L^2 Errors curves

Let us denote by V_{ih} the finite element approximation space on Ω_i, of basis $(\psi_{i,j})_{j=1,\cdots N_i}, i = 1,2$, and by W_h that of $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$, of basis $(\phi_k)_{k=1,\cdots K}$ and $\Lambda := \left\{ \eta \in H^{1/2}(\Gamma) \mid \eta = v_{|_\Gamma} \text{ for a suitable } v \in H^1(\Omega) \right\}$ the trace space. The mortar formulation is given by: for $i = 1,2$ find $u_i \in V_i := H^1(\Omega_i)$, $\lambda \in \Lambda$ such that

$$\begin{cases} \displaystyle\int_{\Omega_i} \nabla u_i \cdot \nabla v_i \pm \int_\Gamma \lambda v_i = \int_{\Omega_i} f v_i & \forall\, v_i \in H^1(\Omega_i) \\ \displaystyle\int_\Gamma \lambda (u_1 - u_2) = 0 & \forall\, \lambda \in \Lambda \end{cases} \tag{5}$$

Listing 3 Jump terms in the global matrix for mortar formulation

```
// product function spaces X_{h_1} × X_{h_2} × Λ_h for Ω_1 × Ω_2 × Γ
typedef meshes<mesh1_type,mesh2_type,trace_mesh_type> mesh_type;
typedef bases<Lagrange<2>,
              Lagrange<3>,
              Lagrange<2,Mortar> > basis_type;
typedef FunctionSpace< mesh_type, basis_type > space_type;
auto mesh = meshes( mesh1, mesh2, trace_mesh );
auto Xh = space_type::New( _mesh=mesh );
auto u = Xh->element();
auto u1 = u.element<0>();
auto u2 = u.element<1>();
```

```
auto mu = u.element<2>();
// assembly of jump terms in the global  matrix A
auto A = M_backend->newMatrix( _trial=Xh, _test=Xh );
form2( _trial=Xh, _test=Xh, _matrix=A ) +=
  integrate(elements(Xh->mesh<3>()),
```

The convergence results in figure 2 are obtained with the solution of the problem (1) using mortar formulation (5) by splitting the initial domain $\Omega = [0,1] \times [0,1]$ into two nonoverlapping subdomains $\Omega_1 = [0,0.45] \times [0,1]$ and $\Omega_2 = [0.45,1] \times [0,1]$ with $g(x,y) = \sin(\pi x)\cos(\pi y)$ is the exact solution and $f(x,y) = 2\pi^2 g$ the right hand side. The convergence tests are performed by taking different mesh sizes $h_{\Omega_1} = h \in \{0.2, 0.1, 0.05, 0.025, 0.0125\}$, $h_{\Omega_2} = h_{\Omega_1} + 10^{-3}$ and different Lagrange polygonal orders $P_k,\ k \in \{1,2,3,4,5\}$. We plot the linear regression lines of $\|u - u_h\|_{L^2}$ versus h, and we retrieve the optimal convergence properties provided by the mortar method. Note that the above 2D/3D mortar code in Listing 3 is purely sequential, the parallel version of 2D/3D mortar code for arbitrary number of subdomains is presented in [18].

4 Conclusion

We presented our ongoing work on building a flexible domain decomposition framework in FEEL++. A lot of work remains to be done, however we have already the toolbox to reproduce a large range of domain decomposition methods in sequential and to a lesser extent in parallel. Regarding the Schwarz methods, we are currently working on having them as preconditioners of Krylov subspace methods and building coarse grid preconditioners on massively parallel architectures, see [9]. As to the mortar methods, we have already a 2D/3D parallel code with some simple preconditioner strategy [18] and we develop scalable preconditioners for the constraint space formulation, see [4].

Acknowledgements The authors would like to thank Frédéric Nataf, Silvia Bertoluzza and Pierre Jolivet for many fruitful discussions. This ongoing work has been sponsored by ANR-Cosinus-HAMM and the Region Rhone-Alpes. It was granted access to the HPC resources of TGCC@CEA made available within the Distributed European Computing Initiative by the PRACE-2IP, receiving funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement RI-283493.

References

1. Bagheri, B., Scott, R.: Analysa. http://people.cs.uchicago.edu/ ridg/al/aa.ps
2. Bangerth, W., Hartmann, R., Kanschat, G.: `deal.II` Differential Equations Analysis Library, Technical Reference. URL `http://www.dealii.org`. `http://www.dealii.org`

3. Belgacem, F.B., Maday, Y.: The mortar element method for three-dimensional finite elements. R.A.I.R.O. Modl. Math. Anal. **31**, 289–302 (1997)
4. Bertoluzza, S., Pennacchio, M.: Analysis of substructuring preconditioners for mortar methods in an abstract framework. Applied Mathematics Letters **20**(2), 131 – 137 (2007). DOI 10.1016/j.aml.2006.02.029. URL `http://www.sciencedirect.com/science/article/pii/S0893965906001108`
5. Chabannes, V., Prud'homme, C., Pena, G.: High order fluid structure interaction in 2D and 3D: Application to blood flow in arteries. Journal of Computational and Applied Mathematics (Accepted) (2012)
6. Del Pino, S., Pironneau, O.: FreeFEM3D Manual. Laboratoire Jacques Louis Lions (2005)
7. Doyeux, V., Chabannes, V., Prud'homme, C., Ismail, M.: Simulation of two fluid flow using a level set method application to bubbles and vesicle dynamics. Journal of Computational and Applied Mathematics (Accepted) (2012)
8. Dular, P., Geuzaine, C.: Getdp: a general environment for the treatment of discrete problems. http://www.geuz.org/getdp
9. Jolivet, P., Dolean, V., Hecht, F., Nataf, F., Prud'homme, C., Spillane, N.: High performance domain decomposition methods on massively parallel architectures with FreeFem++. Journal of Numerical Mathematics **20**(3-4), 287–302 (2013). DOI: 10.1515/jnum-2012-0015
10. Kirby, R.C., Logg, A.: Efficient compilation of a class of variational forms. ACM Trans. Math. Softw. **33**(3), 17 (2007). DOI http://doi.acm.org/10.1145/1268769.1268771
11. Long, K.: Sundance: Rapid development of high-performance parallel finite-element solutions of partial differential equations. http://software.sandia.gov/sundance/
12. Nitsche, J.: ber ein variationsprinzip zur lsung von dirichlet-problemen bei verwendung von teilrumen, die keinen randbedingungen unterworfen sind. Abh. Math. Sem. Univ. Hamburg **36**, 9–15 (1971). Collection of articles dedicated to Lothar Collatz on his sixtieth birthday
13. Pena, G., Prud'homme, C., Quarteroni, A.: High order methods for the approximation of the incompressible navierstokes equations in a moving domain. Computer Methods in Applied Mechanics and Engineering **209-212**, 197–211 (2011)
14. Prud'homme, C.: Life: Overview of a unified C++ implementation of the finite and spectral element methods in 1d, 2d and 3d. In: In Workshop On State-Of-The-Art In Scientific And Parallel Computing, Lecture Notes in Computer Science, page 10. Springer-Verlag (2007)
15. Prud'Homme, C., Chabannes, V., Doyeux, V., Ismail, M., Samake, A., Pena, G., et al.: Feel++: A computational framework for galerkin methods and advanced numerical methods. Esaim Proceedings (2012). URL `http://hal.archives-ouvertes.fr/docs/00/66/35/18/PDF/feel.pdf`. Accepted
16. Quarteroni, A., Valli, A.: Domain decomposition methods for partial Differential equations, chap. The Mathematical Fundation of Domain Decomposition Methods, pp. 1–39. Numerical Mathematics and Scientific Computation. Oxford University Press, New York (1999)
17. Renard, Y., Pommier, J.: Getfem++: Generic and efficient c++ library for finite element methods elementary computations. http://www-gmm.insa-toulouse.fr/getfem/
18. Samake A. Prud'Homme, C., Zaza, C., Chabannes, V.: Parallel implementation of the mortar element method. Esaim Proceedings (2013). In progress
19. Smith, B., Bjorstad, P., Gropp, W.: Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press (2004). URL `http://books.google.fr/books?id=dxwRLu1dBioC`
20. Toselli, A., Widlund, O.: Domain Decomposition Methods - Algorithms and Theory. Springer Series in Computational Mathematics. Springer (2004). URL `http://books.google.fr/books?id=tpSPx68R3KwC`

Additive Schwarz Method for DG Discretization of Anisotropic Elliptic Problems

Maksymilian Dryja[1], Piotr Krzyżanowski[1], and Marcus Sarkis[2]

1 Introduction

In the paper we consider a second order elliptic problem with discontinuous anisotropic coefficients defined on a polygonal region Ω. The problem is discretized by a Discontinuous Galerkin (DG) finite element method with triangular elements and piecewise linear functions. Our goal is to design and analyze an additive Schwarz method (ASM), see the book by Toselli and Widlund [4], for solving the resulting discrete problem with rate of convergence independent of the jumps of the coefficients. The method is two-level and without overlap of Ω_l, the substructures into which the original region Ω is partitioned. It is proved that the convergence of the method is independent of the jumps of the coefficients appearing on triangles inside of Ω_l, see [3]. It is the same for the jumps appearing on triangles which touch $\partial\Omega_l$ under additional assumptions on the coefficients, like monotonicity or quasi-monotonicity. The ASM discussed here is a generalization of method presented in [1]. Numerical experiments confirm the theoretical results.

The paper is organized as follows. In Section 2, differential and discrete DG problems are formulated. In Section 3, ASM for solving the discrete problem is designed and analyzed. Numerical experiments are presented in Section 4.

2 Differential and discrete DG problems

We consider the following elliptic problem: find $u^* \in H_0^1(\Omega)$ such that

$$a(u^*, v) = f(v), \qquad \forall v \in H_0^1(\Omega) \tag{1}$$

where

$$a(u,v) = \int_\Omega \rho(x)\nabla u \cdot \nabla v\, dx, \qquad f(v) = \int_\Omega f v\, dx,$$

$$\rho(x) = \begin{pmatrix} \rho_{11}(x) & \rho_{12}(x) \\ \rho_{21}(x) & \rho_{22}(x) \end{pmatrix}.$$

[1] University of Warsaw, Poland; e-mail: {m.dryja,p.krzyzanowski}@mimuw.edu.pl .[2] Worcester Polytechnic Institute, USA and Instituto Nacional de Matemática Pura e Aplicada, Brasil; e-mail: msarkis@wpi.edu

We assume that Ω is a polygonal region, $f \in L^2(\Omega)$ and $\rho(x)$, the diffusivity tensor, is a symmetric matrix, uniformly positive definite with respect to x, and $\rho_{ij} \in L^\infty(\Omega), i.j = 1,2$. Under these assumptions problem (1) is well posed.

Let $\mathscr{T}^h(\Omega)$ be a triangulation of Ω with triangular elements K_i and the mesh parameter h. We assume that $\mathscr{T}^h(\Omega)$ is shape regular and quasiuniform. Let $X_i(K_i)$ denote a space of linear functions on K_i and

$$X_h(\Omega) = \Pi_{i=1}^N X_i(K_i), \quad \bar{\Omega} = \bigcup_{i=1}^N K_i$$

be the space in which problem (1) is approximated. Note that $X_h(\Omega) \not\subset H^1(\Omega)$ and its elements do not vanish on $\partial\Omega$, in general.

The discrete problem for (1) is of the form: find $u_h^* \in X_h(\Omega)$ such that

$$\hat{a}_h(u_h^*, v_h) = f(v_h), \qquad v_h \in X_h(\Omega), \tag{2}$$

where for $u, v \in X_h(\Omega), u = \{u_i\}_{i=1}^N, u_i \in X_i(K_i)$,

$$\hat{a}_h(u,v) = \sum_{i=1}^N \hat{a}_i(u,v), \qquad f(v) = \sum_{i=1}^N \int_{K_i} f v_i \, dx$$

and

$$\rho^{(i)} = \rho_{|K_i}, \qquad \rho^{(i)} = \{\rho_{kl}^{(i)}\}_{k,l=1}^2,$$

and $\rho_{kl}^{(i)}$ are constants on K_i which can always be assumed for linear elements. Here

$$\hat{a}_i(u,v) = a_i(u,v) + s_i(u,v) + p_i(u,v),$$

with symmetric forms

$$a_i(u,v) = \int_{K_i} \rho^{(i)} \nabla u_i \cdot \nabla v_i \, dx,$$

$$s_i(u,v) = \sum_{E_{ij} \subset \partial K_i} \int_{E_{ij}} \omega_i [n_i^T \rho^{(i)} \nabla u_i (v_j - v_i) + n_i^T \rho^{(i)} \nabla v_i (u_j - u_i)] \, ds,$$

$$p_i(u,v) = \sum_{E_{ij} \subset \partial K_i} \frac{\sigma}{h} \int_{E_{ij}} \gamma_{ij} (u_i - u_j)(v_i - v_j) \, ds$$

where $E_{ij} = E_{ji} = \partial K_i \cap \partial K_j, E_{ij} \subset \partial K_i$ and $E_{ji} \subset \partial K_j$; $n_i = n_{E_{ij}}$ is the unit normal vector to E_{ij} pointing from K_i to K_j;

$$\omega_i \equiv \omega_{E_{ij}} = \frac{\delta_{\rho^n}^{(j)}}{\delta_{\rho^n}^{(i)} + \delta_{\rho^n}^{(j)}}, \qquad \omega_j \equiv \omega_{E_{ji}} = \frac{\delta_{\rho^n}^{(i)}}{\delta_{\rho^n}^{(i)} + \rho_{\rho^n}^{(j)}}$$

and

$$\delta_{\rho^n}^{(i)} = n_i^T \rho^{(i)} n_i, \qquad \delta_{\rho^n}^{(j)} = n_j^T \rho^{(j)} n_j;$$

$\gamma_{ij} \equiv \gamma_{E_{ij}} = 2\delta_{\rho^n}^{(i)} \delta_{\rho^n}^{(j)} / (\delta_{\rho^n}^{(j)} + \delta_{\rho^n}^{(i)})$; σ is a positive (sufficiently large, cf. Lemma 1) penalty parameter, which ensures the ellipticity of $\hat{a}_i(\cdot,\cdot)$.

To analyze problem (2) we introduce some auxiliary bilinear forms and a broken norm. Let the elliptic symmetric form $d_h(\cdot,\cdot)$ be defined as

$$d_h(u,v) = \sum_{i=1}^{N} d_i(u,v), \quad d_i(u,v) = a_i(u,v) + p_i(u,v) \tag{3}$$

and let the weighted broken norm in $X_h(\Omega)$ be defined by

$$\| u \|_{1,h}^2 \equiv d_h(u,u) = \sum_{i=1}^{N} \{ \| (\rho^{(i)})^{1/2} \nabla u_i \|_{L^2(K_i)}^2 + \sum_{E_{ij} \subset \partial K_i} \frac{\sigma}{h} \gamma_{ij} \| u_i - u_j \|_{L^2(E_{ij})}^2 \}. \tag{4}$$

Lemma 1. *There exists $\sigma_0 > 0$ such that for $\sigma \geq \sigma_0$ there exist positive constants C_0 and C_1 independent of $\rho^{(i)}$ and h such tha*

$$C_0 d_i(u,u) \leq \hat{a}_i(u,u) \leq C_1 d_i(u,u)$$

and

$$C_0 d_h(u,u) \leq \hat{a}(u,u) \leq C_1 d_h(u,u)$$

for all $u \in X_h$.

For the proof we refer for example to [1] for isotropic cases and [2] for anisotropic cases.

Lemma 1 implies that the discrete problem (2) is well posed if the penalty parameter $\sigma \geq \sigma_0$. Below σ is fixed and assumed to satisfy the above condition.

The error bound is given by

Theorem 1. *Let u^* and u_h^* be the solutions of (1) and (2). For $u^*_{|_{K_i}} \in H^2(K_i)$ holds*

$$\| u^* - u_h^* \|_{1,h}^2 \leq M h^2 \sum_{i=1}^{N} \lambda_{max}(\rho^{(i)}) |u^*|_{H^2(K_i)}^2$$

where M is independent of h, u^* and ρ_i; $\lambda_{max}(\rho^{(i)})$ is a maximum eigenvalue of $\rho^{(i)}$.

The proof follows from Lemma 1, for details see for example [2].

3 Additive Schwarz method

We design and analyze ASM for solving problem (2) following to the abstract theory of ASMs, see for example, [4].

3.1 Decomposition of $X_h(\Omega)$

Let

$$\bar{\Omega} = \bigcup_{l=1}^{L} \bar{\Omega}_l, \qquad \Omega_l \cap \Omega_m = \{\emptyset\}, \qquad l \neq m$$

where $\bar{\Omega}_l$ is a union of triangulation elements K_i and $H_l = diam(\Omega_l)$. The decomposition of $X_h(\Omega)$ is

$$X_h(\Omega) = X^{(0)}(\Omega) + X^{(1)}(\Omega) + \ldots + X^{(L)}(\Omega),$$

where for $l = 1, \ldots, L$

$$X^{(l)}(\Omega) = \{v = \{v_i\}_{i=1}^N \in X_h(\Omega) : v_i = 0 \text{ on } K_i \not\subset \Omega_l\}$$

and for $l = 0$

$$V^{(0)}(\Omega) = span\{\phi^{(l)}\}_{l=1}^L$$

with $\phi^{(l)} = 1$ on $\bar{\Omega}_l$ and $\phi^{(l)} = 0$ otherwise.

3.2 Inexact local solvers

For $u^{(l)} = \{u_i^{(l)}\}_{i=1}^N \in X^{(l)}(\Omega)$ and $v^{(l)} = \{v_i^{(l)}\}_{i=1}^N \in X^{(l)}(\Omega), l = 1, \ldots, L$, we define

$$b_l(u^{(l)}, v^{(l)}) = d_h(u^{(l)}, v^{(l)}).$$

The overlap between local subproblems is very small (only through the subdomain interface), reducing communication cost to a level similar to substructuring methods. Instead of solving exact subproblems with form $\hat{a}_h(\cdot,\cdot)$ on subdomains, we solve problems with simplified form $d_h(\cdot,\cdot)$. Note that on $X^{(l)}(\Omega) \times X^{(l)}(\Omega)$

$$d_h(u^{(l)}, v^{(l)}) = \sum_{K_i \subset \bar{\Omega}_l} \{(\rho^{(i)} \nabla u_i^{(l)}, \nabla v_i^{(l)})_{L^2(K_i)} + \sum_{E_{ij} \subset \partial K_i} \frac{\sigma}{h} \gamma_{ij} (u_i^{(l)} - u_j^{(l)}, v_i^{(l)} - v_j^{(l)})_{L^2(E_{ij})}\}.$$

For $l = 0$ and $u^{(0)} = \{u_i^{(0)}\}_{i=1}^N \in X^{(0)}(\Omega)$ and $v^{(0)} = \{v_i^{(0)}\}_{i=1}^N \in X^{(0)}(\Omega)$ we set

$$b_0(u^{(0)}, v^{(0)}) = d_h(u^{(0)}, v^{(0)}) \equiv \sum_{l=1}^{L} \frac{\sigma}{h} \sum_{E_{ij} \subset \partial \Omega_l} \gamma_{ij} (u_i^{(0)} - u_j^{(0)}, v_i^{(0)} - v_j^{(0)})_{L^2(E_{ij})}.$$

3.3 Operator equation

For $l = 0, \ldots, L$, let us define $T_l : X_h(\Omega) \to X^{(l)}(\Omega)$ by

$$b_l(T_l u, v) = \hat{a}_h(u, v), \quad v \in X^{(l)}(\Omega).$$

Then problem (2) is replaced by

$$Tu_h^* = g_h, \qquad g_h = \sum_{l=0}^{L} g_l, \qquad g_l = T_l u_h^*. \tag{5}$$

with $T = T_0 + T_1 + \ldots + T_L$. Note that in order to compute g_l we do not need to know u_h^*. From the theorem below it follows that problems (2) and (5) have the same unique solution.

3.4 Analysis

Let $\bar{\Omega}_l^h$ denote a layer around $\partial\Omega_l$. It is a union of $K_i \subset \bar{\Omega}_l$ which touch $\partial\Omega_l$ by edge or/and vertex.
Let

$$\bar{\alpha}_l := \max_{K_i \subset \bar{\Omega}_l^h} \lambda_{\max}(\rho^{(i)}), \qquad \underline{\alpha}_l := \min_{K_i \subset \bar{\Omega}_l^h} \lambda_{\min}(\rho^{(i)})$$

where $\lambda_{\max}(\rho^{(i)})$ and $\lambda_{\min}(\rho^{(i)})$ are maximum and minimum eigenvalues of $\rho^{(i)}$ on K_i.

Theorem 2 (main result). *For any $u \in X_h(\Omega)$ there holds*

$$C_2 \beta^{-1} \hat{a}_h(u,u) \le \hat{a}_h(Tu,u) \le C_3 \hat{a}_h(u,u) \tag{6}$$

where

$$\beta = \max_{1 \le l \le L} \frac{\bar{\alpha}_l H_l^2}{\underline{\alpha}_l h^2}$$

and C_2 and C_3 are positive constants independent of $\rho^{(i)}, \bar{\alpha}_l$ and $\underline{\alpha}_l$ for $i = 1, \ldots, N$ and $l = 1, \ldots, L$.

To prove Theorem 2 we need to check three key assumptions of the abstract theory of ASMs, see Toselli and Widlund book [4]. The proof is omitted here due to the limit of pages and will be published elsewhere.

Remark 1. Note that the convergence of the method is independent of the jumps of $\rho^{(i)}$ on $\bar{\Omega}_l \setminus \Omega_l^h$ for all $l = 1, \ldots, L$, i.e. of the jumps of $\rho^{(i)}$ on K_i which do not touch $\partial\Omega_l$.

Remark 2. Let us mention several specific cases when the above estimate can be improved. When ρ is isotropic and subdomainwise constant, then we can prove that $\beta = \max_l (H_l/h)$ in (6). When $\bar{\alpha}_l$ and $\underline{\alpha}_l$ are the same order and $\underline{\alpha}_l \le \max_{K_i \subset \bar{\Omega}_l} \lambda_{\min}(\rho^{(i)})$,

then $\beta = \max_l(H_l/h)$, i.e. the convergence is independent of the jumps of $\rho^{(i)}$. Estimate (6) can be also improved in the case when $\lambda_{\max}(\rho^{(i)})$ on K_i which touch $\partial\Omega_l$ by edges are monotonic or quasi-monotonic on $\partial\Omega_l$ for $l = 1,\ldots,L$.

4 Numerical experiments

Let us choose the unit square as the domain Ω and for some prescribed integer m divide it into $L = 2^m \times 2^m$ smaller squares Ω_l ($l = 1,\ldots,L$) of equal size. This decomposition of Ω is then further refined into a uniform triangulation $\mathscr{T}^h(\Omega)$ based on a square $2^M \times 2^M$ grid ($M \geq m$) with each square split into two triangles of identical shape. Hence, the fine mesh parameter $h = 2^{-M}$, while the coarse grid parameter is $H = 2^{-m}$. We discretize system (1) on the fine triangulation using method (2) with $\sigma = 7$.

In tables below we report the number of Preconditioned Conjugate Gradient iterations for operator T (defined in Section 3.3) which are required to reduce the initial Euclidean norm of the residual by a factor of 10^6 and (in parentheses) the condition number estimate for T. We consider two sets of test problems: with either anisotropic or discontinuous coefficients matrix ρ. We will always choose a random vector for the right hand side and a zero as the initial guess.

Discontinuous, elementwise constant isotropic coefficients. Let us consider diffusion coefficient of the form

$$\rho(x) = \rho_{11}(x) \cdot I \tag{7}$$

where ρ_{11} equals 1 on even numbered elements (of fine triangulation) and equals 10^{-2} on odd ones. Table 1 shows the dependence on the ratio between H and h in this case.

Fine (M) $\rightarrow$ $\downarrow$ Coarse (m)	2	3	4	5	6
2	33 (32)	82 (300)	133 (530)	164 (840)	237 (2000)
3		45 (41)	140 (370)	189 (700)	225 (1100)
4			48 (42)	155 (470)	186 (690)
5				41 (48)	155 (470)
6					49 (44)

Table 1 Dependence of the number of iterations and the condition number (in parentheses) on the ratio H/h, where $H = 2^{-m}$ and $h = 2^{-M}$. Isotropic, elementwise constant coefficient.

Next, let us fix the number of subdomains and the fine mesh size so that $M = 3$ and $m = 5$ and thus $H/h = 4$. Table 2 shows the dependence of the convergence rate and the condition number as we vary the value of ρ_{11} on odd-numbered triangles; on even triangles it remains equal to 1 as previously.

ρ_{11}	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
iter (cond)	61 (80)	72 (10^2)	167 ($6\cdot 10^2$)	335 ($5\cdot 10^3$)	485 ($5\cdot 10^4$)	613 ($5\cdot 10^5$)	743 ($5\cdot 10^6$)

Table 2 Dependence of the number of iterations and the condition number (in parentheses) on the discontinuity in the isotropic, elementwise constant coefficient. Fixed $H/h = 4$.

Indeed, the condition number estimates agree well with our theory regarding the dependence on the discontinuity of the coefficient. In our testcase the increase in the condition number is rather linear than quadratic in H/h, as reported in Table 1. This behaviour is in agreement with our Remark 2. Let us also explain that low iteration numbers in Table 2 are due to a very rapid residual in the residual during the initial phase of the iteration.

Discontinuous, domainwise constant isotropic coefficients. Here we consider ρ as in (7), with discontinuities aligned with an auxiliary partitioning of Ω into 4×4 squares. Precisely, we introduce a red–black checkerboard coloring of this partitioning and set $\rho = 1$ in red regions, and the value of ρ_{11} reported in Table 3 in black ones. In this way, our decomposition of the domain with $M = 5$ and $m = 3$ will always be aligned with the discontinuites and Table 3 shows the dependence on ρ_{11} in this case.

ρ_{11}	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
iter (cond)	61 (80)	60 (70)	58 (67)	58 (68)	62 (68)	64 (68)	67 (68)

Table 3 Dependence of the number of iterations and the condition number (in parentheses) on the discontinuity when the coefficient is isotropic and constant inside subdomains. Red–black 4×4 distribution of ρ, aligned with domain decomposition. Fixed $H/h = 4$.

As predicted in Remark 2, there is no dependence on the discontinuity in the coefficients in this case until the coefficient remains continuous (constant) inside subdomain. This behaviour is not observed when the red–black partitioning is not aligned with the subdomains Ω_l: corresponding numbers for a 3×3 partitioning are shown in Table 4.

ρ_{11}	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
iter (cond)	62 (80)	68 (130)	85 (710)	96 ($7\cdot 10^3$)	113 ($7\cdot 10^4$)	126 ($7\cdot 10^5$)	140 ($7\cdot 10^6$)

Table 4 Dependence of the number of iterations and the condition number (in parentheses) on the discontinuity when the coefficient is isotropic and discontinuous across subdomain boundaries. Red–black 3×3 distribution of ρ, not aligned with the domain decomposition. Fixed $H/h = 4$.

Anisotropic, discontinuous coefficients. Let us continue with the 4×4 red–black partitioning and let us set the coefficient matrix ρ equal to ρ^R in red regions and ρ^B in black ones, where

$$\rho^R(x) = \begin{pmatrix} 10+\rho_{22} & 0 \\ 0 & \rho_{22}, \end{pmatrix} \qquad \rho^B(x) = \begin{pmatrix} \rho_{22} & 0 \\ 0 & 10+\rho_{22} \end{pmatrix},$$

with constant ρ_{22} as specified in Table 5. In this way ρ is constant in both red and black regions, but it suffers from discontinuity across the partitioning borders; the jump is always equal to 10, while the anisotropy ratio is $1+10/\rho_{22}$. The condition numbers grow linearly with the growth of ρ_{22}, which agrees with Theorem 2.

ρ_{22}	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
iter (cond)	60 (82)	94 (210)	222 (10^3)	463 (10^4)	680 (10^5)	782 (10^6)	897 (10^7)

Table 5 Dependence on the anisotropy for discontinuous, piecewise constant coefficient. Fixed $H/h = 4$.

Anisotropic, constant coefficients. Finally, let us consider

$$\rho(x) = \begin{pmatrix} 1 & 0 \\ 0 & \rho_{22} \end{pmatrix}$$

with ρ_{22} constant throughout entire Ω, assuming values specified in Table 6.

ρ_{22}	10^0	10^1	10^2	10^3	10^4	10^5	10^6
iter (cond)	60 (82)	74 (10^2)	159 ($6\cdot 10^2$)	159 ($6\cdot 10^2$)	144 ($6\cdot 10^2$)	143 ($6\cdot 10^2$)	124 ($7\cdot 10^2$)

Table 6 Dependence on the anisotropy. Fixed $H/h = 4$. Continuous, constant coefficient.

It turns out that after initial linear increase in the condition number for moderate ρ_{22}, the condition number is insensitive to further growth of the anisotropy ratio ρ_{22}. This observation can also be explained on the ground of our theory; the details will be provided elsewhere.

Acknowledgements The research of the first two authors has been partially supported by the Polish National Science Centre grant 2011/01/B/ST1/01179.

References

1. Dryja, M., Sarkis, M.: Additive average Schwarz methods for discretization of elliptic problems with highly discontinuous coefficients. Comput. Methods Appl. Math. **10**(2), 164–176 (2010)
2. Ern, A., Stephansen, A.F., Zunino, P.: A discontinuous Galerkin method with weighted averages for advection-diffusion equations with locally small and anisotropic diffusivity. IMA J. Numer. Anal. **29**(2), 235–256 (2009). DOI 10.1093/imanum/drm050. URL `http://dx.doi.org/10.1093/imanum/drm050`
3. Graham, I.G., Lechner, P.O., Scheichl, R.: Domain decomposition for multiscale PDEs. Numer. Math. **106**(4), 589–626 (2007). DOI 10.1007/s00211-007-0074-1. URL `http://dx.doi.org/10.1007/s00211-007-0074-1`
4. Toselli, A., Widlund, O.: Domain decomposition methods—algorithms and theory, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin (2005)

A one-level additive Schwarz preconditioner for a discontinuous Petrov-Galerkin method

Andrew T. Barker[1], Susanne C. Brenner[1], Eun-Hee Park[2], and Li-Yeng Sung[1]

1 A discontinuous Petrov-Galerkin method for a model Poisson problem

Discontinuous Petrov-Galerkin (DPG) methods are new discontinuous Galerkin methods [3, 5, 7, 6, 4, 8] with interesting properties. In this article we consider a domain decomposition preconditioner for a DPG method for the Poisson problem.

Let Ω be a polyhedral domain in $\mathbb{R}^d$ ($d = 2,3$), Ω_h be a simplicial triangulation of Ω. Following the notation in [8], the model Poisson problem (in an ultraweak formulation) is to find $\mathbf{u} \in U$ such that

$$b(\mathbf{u},\mathbf{v}) = l(\mathbf{v}) \qquad \forall \mathbf{v} \in V,$$

where $U = [L_2(\Omega)]^d \times L_2(\Omega) \times H_0^{\frac{1}{2}}(\partial\Omega_h) \times H^{-\frac{1}{2}}(\partial\Omega_h)$, $V = H(\mathrm{div};\Omega_h) \times H^1(\Omega_h)$,

$$b(\mathbf{u},\mathbf{v}) = \int_\Omega \sigma\cdot\tau\,dx - \sum_{K\in\Omega_h}\int_K u\,\mathrm{div}\,\tau\,dx + \sum_{K\in\Omega_h}\int_{\partial K}\hat{u}\,\tau\cdot n\,ds$$
$$- \sum_{K\in\Omega_h}\int_K \sigma\cdot\mathrm{grad}\,v\,dx + \sum_{K\in\Omega_h}\int_{\partial K} v\,\hat{\sigma}_n\,ds$$

for $\mathbf{u} = (\sigma,u,\hat{u},\hat{\sigma}_n) \in U$ and $\mathbf{v} = (\tau,v) \in V$, and $l(\mathbf{v}) = \int_\Omega fv\,dx$.

Here $H_0^{1/2}(\partial\Omega_h)$ (resp. $H^{-1/2}(\partial\Omega_h)$) is the subspace of $\prod_{K\in\Omega_h} H^{1/2}(\partial K)$ (resp. $\prod_{K\in\Omega_h} H^{-1/2}(\partial K)$) consisting of the traces of functions in $H_0^1(\Omega)$ (resp. traces of the normal components of vector fields in $H(\mathrm{div};\Omega)$), and $H(\mathrm{div};\Omega_h)$ (resp. $H^1(\Omega_h)$) is the space of piecewise $H(\mathrm{div})$ vector fields (resp. H^1 functions). The inner product on V is given by

$$\big((\tau_1,v_1),(\tau_2,v_2)\big)_V = \sum_{K\in\Omega_h}\int_K [\tau_1\cdot\tau_2 + \mathrm{div}\,\tau_1\mathrm{div}\,\tau_2 + v_1v_2 + \mathrm{grad}\,v_1\cdot\mathrm{grad}\,v_2]\,dx.$$

The DPG method for the Poisson problem computes $\mathbf{u}_h \in U_h$ such that

$$b(\mathbf{u}_h,\mathbf{v}) = l(\mathbf{v}) \qquad \forall \mathbf{v} \in V_h. \tag{1}$$

Here the trial space $U_h\,(\subset U)$ is defined by

[1] Department of Mathematics and Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, USA, e-mail: {andrewb}{brenner}{sung}@math.lsu.edu .[2] Division of Computational Mathematics, National Institute for Mathematical Sciences, Daejeon 305-811, South Korea, e-mail: eunheepark@nims.re.kr

$$U_h = \prod_{K\in\Omega_h} [P_m(K)]^d \times \prod_{K\in\Omega_h} P_m(K) \times \tilde{P}_{m+1}(\partial\Omega_h) \times P_m(\partial\Omega_h),$$

$P_m(K)$ is the space of polynomials of total degree $\leq m$ on an element K, $\tilde{P}_{m+1}(\partial\Omega_h) = H_0^{1/2}(\partial\Omega_h) \cap \prod_{K\in\Omega_h} \tilde{P}_{m+1}(\partial K)$, where $\tilde{P}_{m+1}(\partial K)$ is the restriction of $P_{m+1}(K)$ to ∂K, and $P_m(\partial\Omega_h) = H^{-1/2}(\partial\Omega_h) \cap \prod_{K\in\Omega_h} P_m(\partial K)$, where $P_m(\partial K)$ is the space of piecewise polynomials on the faces of K with total degree $\leq m$.

Let $V^r = \{(\tau, v) \in V : \tau|_K \in [P_{m+2}(K)]^d, v|_K \in P_r(K)\ \forall K \in \Omega_h\}$ for some $r \geq m+d$. The discrete trial-to-test map $T_h : U_h \longrightarrow V^r$ is defined by

$$(T_h \mathbf{u}_h, \mathbf{v})_V = b(\mathbf{u}_h, \mathbf{v}), \quad \forall \mathbf{u}_h \in U_h,\ \mathbf{v} \in V^r,$$

and the test space V_h is $T_h U_h$.

We can rewrite (1) as $a_h(\mathbf{u}_h, \mathbf{w}) = l(T_h \mathbf{w})$ for all $\mathbf{w} \in U_h$, where

$$a_h(\mathbf{u}, \mathbf{w}) = b_h(\mathbf{u}, T_h \mathbf{w}) = (T_h \mathbf{u}, T_h \mathbf{w})_V$$

is an SPD bilinear form on $V_h \times V_h$, and we define an operator $A_h : U_h \longrightarrow U_h'$ by

$$\langle A_h \mathbf{u}, \mathbf{w}\rangle = a_h(\mathbf{u}, \mathbf{w}) \qquad \forall \mathbf{u}, \mathbf{w} \in U_h. \tag{2}$$

Our goal is to develop a one-level additive Schwarz preconditioner for A_h (cf. [9]).

To avoid the proliferation of constants, we will use the notation $A \lesssim B$ (or $B \gtrsim A$) to represent the inequality $A \leq (\text{constant}) \times B$, where the positive constant only depends on the shape regularity of Ω_h and the polynomial degrees m and r. The notation $A \approx B$ is equivalent to $A \lesssim B$ and $B \lesssim A$.

A fundamental result in [8] is the equivalence

$$a_h(\mathbf{u}, \mathbf{u}) \approx \|\sigma\|^2_{L_2(\Omega)} + \|u\|^2_{L_2(\Omega)} + \|\hat{u}\|^2_{H^{1/2}(\partial\Omega_h)} + \|\hat{\sigma}_n\|^2_{H^{-1/2}(\partial\Omega_h)} \tag{3}$$

that holds for all $\mathbf{u} = (\sigma, u, \hat{u}, \hat{\sigma}_n) \in U_h$, where

$$\|\hat{u}\|^2_{H^{1/2}(\partial\Omega_h)} = \sum_{K\in\Omega_h} \|\hat{u}\|^2_{H^{1/2}(\partial K)} = \sum_{K\in\Omega_h} \inf_{w\in H^1(K), w|_{\partial K} = \hat{u}} \|w\|^2_{H^1(K)}, \tag{4}$$

$$\|\hat{\sigma}_n\|^2_{H^{-1/2}(\partial\Omega_h)} = \sum_{K\in\Omega_h} \|\hat{\sigma}_n\|^2_{H^{-1/2}(\partial K)} = \sum_{K\in\Omega_h} \inf_{q\in H(\mathrm{div};K), q\cdot n|_{\partial K} = \hat{\sigma}_n} \|q\|^2_{H(\mathrm{div};K)}. \tag{5}$$

Therefore the analysis of domain decomposition preconditioners for A_h requires a better understanding of the norms $\|\cdot\|_{H^{1/2}(\partial K)}$ and $\|\cdot\|_{H^{-1/2}(\partial K)}$ on the discrete spaces $\tilde{P}_{m+1}(\partial K)$ and $P_m(\partial K)$.

2 Explicit Expressions for the Norms on $\tilde{P}_{m+1}(\partial K)$ and $P_m(\partial K)$

Lemma 1. *We have*

$$\|\tilde{\zeta}\|^2_{H^{1/2}(\partial K)} \approx h_K\Big(\|\tilde{\zeta}\|^2_{L_2(\partial K)} + \sum_{F\in\Sigma_K} |\tilde{\zeta}|^2_{H^1(F)}\Big) \qquad \forall\, \tilde{\zeta}\in \tilde{P}_{m+1}(\partial K),$$

where h_K is the diameter of K and Σ_K is the set of the faces of K.

Proof. Let $\mathscr{N}(K)$ be the set of nodal points of the P_{m+1} Lagrange finite element associated with K and $\mathscr{N}(\partial K)$ be the set of points in $\mathscr{N}(K)$ that are on ∂K.

Given any $\tilde{\zeta}\in\tilde{P}_{m+1}(\partial K)$, we define $\tilde{\zeta}_*\in P_{m+1}(K)$ by

$$\tilde{\zeta}_*(p) = \begin{cases} \tilde{\zeta}(p) & \text{if } p\in\mathscr{N}(\partial K),\\ \tilde{\zeta}_{\partial K} & \text{if } p\in\mathscr{N}(K)\setminus\mathscr{N}(\partial K),\end{cases} \tag{6}$$

where $\tilde{\zeta}_{\partial K}$ is the mean value of $\tilde{\zeta}$ over ∂K. Since $\tilde{\zeta}_* = \tilde{\zeta}$ on ∂K, we have

$$\|\tilde{\zeta}\|_{H^{1/2}(\partial K)} = \inf_{w\in H^1(K), w|_{\partial K}=\tilde{\zeta}} \|w\|_{H^1(K)} \le \|\tilde{\zeta}_*\|_{H^1(K)}. \tag{7}$$

Suppose $w\in H^1(K)$ satisfies $w=\tilde{\zeta}$ on ∂K. It follows from (6) and the trace theorem with scaling that

$$\|\tilde{\zeta}_*\|^2_{L_2(K)} \lesssim h_K\|\tilde{\zeta}\|^2_{L_2(\partial K)} = h_K\|w\|^2_{L_2(\partial K)} \lesssim \|w\|^2_{H^1(K)}, \tag{8}$$

and, by standard estimates,

$$\begin{aligned} |\tilde{\zeta}_*|^2_{H^1(K)} = |\tilde{\zeta}_* - \tilde{\zeta}_{\partial K}|^2_{H^1(K)} &\lesssim h_K^{-1}\|\tilde{\zeta}_* - \tilde{\zeta}_{\partial K}\|^2_{L_2(\partial K)}\\ &= h_K^{-1}\|w - w_{\partial K}\|^2_{L_2(\partial K)} \lesssim |w|^2_{H^1(K)}. \end{aligned} \tag{9}$$

Combining (7)–(9), we have $\|\tilde{\zeta}\|^2_{H^{1/2}(\partial K)} \approx \|\tilde{\zeta}_*\|^2_{H^1(K)}$. The lemma then follows from (6), the equivalence of norms on finite dimensional spaces and scaling. □

Lemma 2. *We have*

$$\|\zeta\|^2_{H^{-1/2}(\partial K)} \approx h_K\|\zeta\|^2_{L_2(\partial K)} + h_K^{-d}\Big(\int_{\partial K}\zeta ds\Big)^2 \qquad \forall\,\zeta\in P_m(\partial K).$$

Proof. We begin with the reference simplex $\hat{K}$. Let $RT_m(\hat{K})$ be the m-th order Raviart-Thomas space (cf. [2]). Given any $\zeta\in P_m(\partial\hat{K})$, we introduce a (nonempty) subspace $RT_m(\hat{K},\zeta) = \{q\in RT_m(\hat{K}) : q\cdot n = \zeta \text{ on } \partial\hat{K} \text{ and } \operatorname{div} q\in P_0(\hat{K})\}$ of $RT_m(\hat{K})$.

Let $\zeta_*\in RT_m(\hat{K},\zeta)$ be defined by

$$\zeta_* = \min_{q\in RT_m(\hat{K},\zeta)} \|q\|_{L_2(\hat{K})}.$$

Then the map $\hat{S}: P_m(\partial\hat{K}) \longrightarrow RT_m(\hat{K})$ that maps ζ to ζ_* is linear and one-to-one, and we have $(\hat{S}\zeta)\cdot n = \zeta$ on $\partial\hat{K}$, $\operatorname{div}(\hat{S}\zeta)\in P_0(\hat{K})$ and

$$\|\hat{S}\zeta\|_{L_2(\hat{K})} \approx \|\zeta\|_{L_2(\partial\hat{K})} \qquad \forall\,\zeta\in P_m(\partial\hat{K}). \tag{10}$$

Let $\zeta_1,\dots,\zeta_{N_m}$ be a basis of $P_m(\partial\hat{K})$ and $1=\phi_1,\dots,\phi_{N_m}\in H^{1/2}(\partial\hat{K})$ satisfy $\det\left[\int_{\partial\hat{K}}\zeta_i\phi_j\,d\hat{s}\right]_{1\le i,j\le N_m}\neq 0$. We define the map $\hat{Q}:H(\mathrm{div};\hat{K})\longrightarrow P_m(\partial\hat{K})$ by

$$\int_{\partial\hat{K}}(\hat{Q}q)\phi_j\,d\hat{s}=\langle q\cdot n,\phi_j\rangle_{H^{-1/2}(\partial\hat{K})\times H^{1/2}(\partial\hat{K})}\qquad\text{for}\quad 1\le j\le N_m.$$

It follows from the definition of $\hat{Q}$ that $\|\hat{Q}q\|_{L_2(\partial\hat{K})}\lesssim\|q\|_{H(\mathrm{div};\hat{K})}$ for all $q\in H(\mathrm{div};\hat{K})$, and $\hat{Q}q=\zeta$ if $q\cdot n=\zeta\in P_m(\partial\hat{K})$, in which case

$$\|\hat{S}\zeta\|_{L_2(\hat{K})}\lesssim\|\zeta\|_{L_2(\partial\hat{K})}=\|\hat{Q}q\|_{L_2(\partial\hat{K})}\lesssim\|q\|_{H(\mathrm{div};\hat{K})}.\tag{11}$$

Moreover, since $\phi_1=1$, we have

$$\int_{\hat{K}}\mathrm{div}\,(\hat{S}\zeta)\,d\hat{x}=\int_{\partial\hat{K}}(\hat{Q}q)1\,d\hat{s}=\langle q\cdot n,1\rangle_{H^{-1/2}(\partial\hat{K})\times H^{1/2}(\partial\hat{K})}=\int_{\hat{K}}\mathrm{div}\,q\,d\hat{x}$$

and hence

$$\|\mathrm{div}\,(\hat{S}\zeta)\|_{L_2(\hat{K})}\lesssim\|\mathrm{div}\,q\|_{L_2(\hat{K})}.\tag{12}$$

Now we turn to a general simplex K. It follows from (10)–(12) and standard properties of the Piola transform for $H(\mathrm{div})$ (cf. [10]) that there exists a linear map $S:P_m(\partial K)\longrightarrow RT_m(K)$ with the following properties:
(i) $(S\zeta)\cdot n=\zeta$ and hence

$$\|\zeta\|_{H^{-1/2}(\partial K)}=\inf_{q\in H(\mathrm{div};K),\,q\cdot n|_{\partial K}=\zeta}\|q\|_{H(\mathrm{div};K)}\le\|S\zeta\|_{H(\mathrm{div};K)}\qquad\forall\,\zeta\in P_m(\partial K),$$

(ii) for any $q\in H(\mathrm{div};K)$ such that $q\cdot n=\zeta$, we have

$$\|S\zeta\|_{H(\mathrm{div};K)}\lesssim\|q\|_{H(\mathrm{div};K)},$$

(iii) $\mathrm{div}\,(S\zeta)\in P_0(K)$ and hence

$$\int_K\mathrm{div}\,(S\zeta)\,dx=\int_{\partial K}\zeta\,ds\quad\text{or}\quad\|\mathrm{div}\,(S\zeta)\|^2_{L_2(K)}=\Big(\int_{\partial K}\zeta\,ds\Big)^2/|K|,$$

(iv) we have

$$h_K^{-d}\|S\zeta\|^2_{L_2(K)}\approx h_K^{-(d-1)}\|\zeta\|^2_{L_2(\partial K)}.$$

Properties (i)–(iv) then imply

$$\|\zeta\|^2_{H^{-1/2}(\partial K)}\approx\|S\zeta\|^2_{H(\mathrm{div};K)}\approx h_K\|\zeta\|^2_{L_2(\partial K)}+h_K^{-d}\Big(\int_{\partial K}\zeta\,ds\Big)^2.\qquad\square$$

3 A Domain Decomposition Preconditioner

Let Ω be partitioned into overlapping subdomains $\Omega_1,\dots,\Omega_J$ that are aligned with Ω_h. The overlap among the subdomains is measured by δ and we assume (cf. [11]) there is a partition of unity $\theta_1,\dots,\theta_J \in C^\infty(\bar{\Omega})$ that satisfies the usual properties: $\theta_j \geq 0$, $\sum_{j=1}^J \theta_j = 1$ on $\bar{\Omega}$, $\theta_j = 0$ on $\Omega \setminus \Omega_j$, and

$$\|\nabla\theta_j\|_{L_\infty(\Omega)} \lesssim \delta^{-1} \qquad \forall\, 1 \leq j \leq J. \tag{13}$$

We take the subdomain space to be $U_j = \{\mathfrak{u} \in U_h : \mathfrak{u} = 0 \text{ on } \Omega \setminus \Omega_j\}$. Let $\mathfrak{u} = (\sigma, u, \hat{u}, \hat{\sigma}_n) \in U_h$. Then $\mathfrak{u} \in U_j$ if and only if (i) σ and u vanish on every K outside Ω_j and (ii) $\hat{u}$ and $\hat{\sigma}_n$ vanish on ∂K for every K outside Ω_j. We define $a_j(\cdot,\cdot)$ to be the restriction of $a_h(\cdot,\cdot)$ on $U_j \times U_j$. Let $A_j : U_j \longrightarrow U_j'$ be defined by

$$\langle A_j\mathfrak{u}_j, \mathfrak{w}_j\rangle = a_j(\mathfrak{u}_j, \mathfrak{w}_j) \qquad \forall\, \mathfrak{u}_j, \mathfrak{w}_j \in U_j. \tag{14}$$

It follows from (3) that

$$a_j(\mathfrak{u}_j,\mathfrak{u}_j) \approx \|\sigma_j\|^2_{L_2(\Omega_j)} + \|u_j\|^2_{L_2(\Omega_j)} + \|\hat{u}_j\|^2_{H^{1/2}(\partial\Omega_{j,h})} + \|\hat{\sigma}_{n,j}\|^2_{H^{-1/2}(\partial\Omega_{j,h})}, \tag{15}$$

where $\mathfrak{u}_j = (\sigma_j, u_j, \hat{u}_j, \hat{\sigma}_{n,j}) \in U_j$, $\Omega_{j,h}$ is the triangulation of Ω_j induced by Ω_h and the norms $\|\cdot\|_{H^{1/2}(\partial\Omega_{j,h})}$ and $\|\cdot\|_{H^{-1/2}(\partial\Omega_{j,h})}$ are analogous to those in (4) and (5).

Let $I_j : U_j \longrightarrow U_h$ be the natural injection. The one-level additive Schwarz preconditioner $B_h : U_h' \longrightarrow U_h$ is defined by

$$B_h = \sum_{j=1}^J I_j A_j^{-1} I_j^t.$$

Lemma 3. *We have*

$$\lambda_{\min}(B_hA_h) \gtrsim \delta^2.$$

Proof. Let $I_{h,1}$, $I_{h,2}$, $I_{h,3}$ and $I_{h,4}$ be the nodal interpolation operators for the components $\prod_{K\in\Omega_h}\big[P_m(K)\big]^d$, $\prod_{K\in\Omega_h} P_m(K)$, $\tilde{P}_{m+1}(\partial\Omega_h)$ and $P_m(\partial\Omega_h)$ of U_h respectively. Given any $\mathfrak{u} = (\sigma, u, \hat{u}, \hat{\sigma}_n) \in U_h$, we define $\mathfrak{u}_j \in U_j$ by

$$\mathfrak{u}_j = \big(I_{h,1}(\theta_j\sigma), I_{h,2}(\theta_j u), I_{h,3}(\theta_j\hat{u}), I_{h,4}(\theta_j\hat{\sigma}_n)\big).$$

Then we have $\mathfrak{u} = \sum_{j=1}^J \mathfrak{u}_j$ and, in view of (14) and (15),

$$\begin{aligned}\langle A_j\mathfrak{u}_j,\mathfrak{u}_j\rangle \approx{}& \|I_{h,1}(\theta_j\sigma)\|^2_{L_2(\Omega_j)} + \|I_{h,2}(\theta_j u)\|^2_{L_2(\Omega_j)} \\ &+ \|I_{h,3}(\theta_j\hat{u})\|^2_{H^{1/2}(\partial\Omega_{j,h})} + \|I_{h,4}(\theta_j\hat{\sigma}_n)\|^2_{H^{-1/2}(\partial\Omega_{j,h})}.\end{aligned} \tag{16}$$

The following bounds for the first two terms on the right-hand side of (16) are straightforward:

$$\|I_{h,1}(\theta_j\sigma)\|^2_{L_2(\Omega_j)} \lesssim \|\sigma\|^2_{L_2(\Omega_j)} \quad \text{and} \quad \|I_{h,2}(\theta_j u)\|^2_{L_2(\Omega_j)} \lesssim \|u\|^2_{L_2(\Omega_j)}. \tag{17}$$

We will use Lemma 1 and Lemma 2 to derive the following bounds

$$\|I_{h,3}(\theta_j\hat{u})\|^2_{H^{1/2}(\partial\Omega_{j,h})} \lesssim \delta^{-2}\|\hat{u}\|^2_{H^{1/2}(\partial\Omega_{j,h})}, \tag{18}$$

$$\|I_{h,4}(\theta_j\hat{\sigma}_n)\|^2_{H^{-1/2}(\partial\Omega_{j,h})} \lesssim \delta^{-2}\|\hat{\sigma}_n\|^2_{H^{-1/2}(\partial\Omega_{j,h})}. \tag{19}$$

Let $K \in \Omega_{j,h}$. It follows from Lemma 1, (13) and standard discrete estimates that

$$\begin{aligned}
\|I_{h,3}(\theta_j\hat{u})\|^2_{H^{1/2}(\partial K)} &\approx h_K\Big(\|I_{h,3}(\theta_j\hat{u})\|^2_{L_2(\partial K)} + \sum_{F\in\Sigma_K}|I_{h,3}(\theta_j\hat{u})|^2_{H^1(F)}\Big)\\
&\lesssim h_K\|\hat{u}\|^2_{L_2(\partial K)} + h_K\sum_{F\in\Sigma_K}\big(\|\nabla\theta_j\|^2_{L_\infty(\Omega)}\|\hat{u}\|^2_{L_2(F)} + \|\theta_j\|^2_{L_\infty(\Omega)}|\hat{u}|^2_{H^1(F)}\big)\\
&\lesssim h_K\|\hat{u}\|^2_{L_2(\partial K)} + h_K\delta^{-2}\|\hat{u}\|^2_{L_2(\partial K)} + h_K\sum_{F\in\Sigma_K}|\hat{u}|^2_{H^1(F)} \lesssim \delta^{-2}\|\hat{u}\|^2_{H^{1/2}(\partial K)}.
\end{aligned}$$

Summing up this estimate over all the simplexes in $\Omega_{j,h}$ yields (18).

Similarly, it follows from Lemma 2 and (13) that

$$\begin{aligned}
\|I_{h,4}(\theta_j\hat{\sigma}_n)\|^2_{H^{-1/2}(\partial\hat{K})} &\approx h_K\|I_{h,4}(\theta_j\hat{\sigma}_n)\|^2_{L_2(\partial K)} + h_K^{-d}\Big(\int_{\partial K} I_{h,4}(\theta_j\hat{\sigma}_n)\,ds\Big)^2\\
&\lesssim h_K\|\hat{\sigma}_n\|^2_{L_2(\partial K)} + h_K^{-d}\Big(\int_{\partial K} I_{h,4}\big[(\theta_j-\theta_j^K)\hat{\sigma}_n\big]\,ds\Big)^2 + h_K^{-d}(\theta_j^K)^2\Big(\int_{\partial K}\hat{\sigma}_n\,ds\Big)^2\\
&\lesssim h_K\|\hat{\sigma}_n\|^2_{L_2(\partial K)} + h_K\delta^{-2}\|\hat{\sigma}_n\|^2_{L_2(\partial K)} + h_K^{-d}\Big(\int_{\partial K}\hat{\sigma}_n\,ds\Big)^2 \lesssim \delta^{-2}\|\hat{\sigma}_n\|^2_{H^{-1/2}(\partial K)},
\end{aligned}$$

where θ_j^K is the mean value of σ_j over K. Summing up this estimate over all the simplexes in $\Omega_{j,h}$ gives us (19).

Putting (2), (3) and (16)–(19) together we find $\sum_{j=1}^J\langle A_j\mathfrak{u}_j,\mathfrak{u}_j\rangle \lesssim \delta^{-2}\langle A_h\mathbf{u},\mathbf{u}\rangle$, which implies $\lambda_{\min}(B_hA_h) \gtrsim \delta^2$ by the standard theory of additive Schwarz preconditioners [11]. □

Combining Lemma 3 with the standard estimate $\lambda_{\max}(B_hA_h) \lesssim 1$, we obtain the following theorem.

Theorem 1. *We have*

$$\kappa(B_hA_h) = \frac{\lambda_{\max}(B_hA_h)}{\lambda_{\min}(B_hA_h)} \leq C\delta^{-2},$$

where the positive constant C depends only on the shape regularity of Ω_h and the polynomial degrees m and r.

Remark 1. Theorem 1 is also valid for DPG methods based on tensor product finite elements.

4 Numerical results

We solve the Poisson problem on the square $(0,1)^2$ with exact solution $u = \sin(\pi x_1)\sin(\pi x_2)$ and uniform square meshes. The trial space is based on Q_1 polynomials for σ and u, P_2 polynomials for $\hat{u}$, and P_1 polynomials for $\hat{\sigma}_n$. We use bicubic polynomials for the space V^r in the construction of the trial-to-test map T_h.

The number of conjugate gradient iterations required to reduce the residual by 10^{10} are given in Table 1 for four overlapping subdomains. The linear growth of the number of iterations for the unpreconditioned system is consistent with the condition number estimate $\kappa(A_h) \lesssim h^{-2}$ in [8]. Note that in this case the boundary of every subdomain has a nonempty intersection with $\partial\Omega$ and it is not difficult to use a discrete Poincaré inequality to show that the estimate in Theorem 1 can be improved to $\kappa(B_hA_h) \lesssim |\ln h|\delta^{-1}$. This is consistent with the observed growth of the number of iterations for the preconditioned system as δ decreases.

Table 1 Number of iterations for the Schwarz preconditioner with subdomain size $H = 1/2$.

h	δ	unpreconditioned	preconditioned
2^{-2}	2^{-2}	496	14
2^{-3}	2^{-3}	1556	17
	2^{-2}		14
2^{-4}	2^{-4}	3865	20
	2^{-3}		17
	2^{-2}		14
2^{-5}	2^{-5}	8793	27
	2^{-4}		20
	2^{-3}		18

In Table 2 we display the results for $h = 2^{-5}$ and various subdomain sizes H with $\delta = H/2$. The estimate $\kappa(B_hA_h) \lesssim \delta^{-2} \approx H^{-2}$ is consistent with the observed linear growth of the number of iterations for the preconditioned system as H decreases. Such a condition number estimate for the one-level additive Schwarz preconditioner is known to be sharp for standard finite element methods [1].

Table 2 Number of iterations with $h = 2^{-5}$ and various subdomain sizes H with $\delta = H/2$.

h	H	unpreconditioned	preconditioned
2^{-5}	2^{-1}	8793	15
	2^{-2}		25
	2^{-3}		45
	2^{-4}		89

Acknowledgements The work of the first author was supported in part by the National Science Foundation VIGRE Grant DMS-07-39382. The work of the second and fourth authors was supported in part by the National Science Foundation under Grant No. DMS-10-16332. The work of the third author was supported in part by a KRCF research fellowship for young scientists. The authors would also like to thank Leszek Demkowicz for helpful discussions.

References

1. Brenner, S.C.: Lower bounds in domain decomposition. In: Domain Decomposition Methods in Science and Engineering XVI, pp. 27–39. Springer, Berlin (2007)
2. Brezzi, F., Fortin, M.: Mixed and Hybrid Finite Element Methods. Springer-Verlag, New York-Berlin-Heidelberg (1991)
3. Demkowicz, L., Gopalakrishnan, J.: A class of discontinuous Petrov–Galerkin methods. Part I: The transport equation. Comp. Meth. Appl. Math. Engrg. **199**, 1558–1572 (2010)
4. Demkowicz, L., Gopalakrishnan, J.: Analysis of the DPG method for the Poisson equation. SIAM J. Numer. Anal. **49**, 1788–1809 (2011)
5. Demkowicz, L., Gopalakrishnan, J.: A class of discontinuous Petrov–Galerkin methods. Part II: Optimal test functions. Num. Meth. Part. Diff. Eq. **27**, 70–105 (2011)
6. Demkowicz, L., Gopalakrishnan, J.: A class of discontinuous Petrov–Galerkin methods. Part IV: The optimal test norm and time–harmonic wave propagation in 1D. J. Comp. Phys. **230**, 2406–2432 (2011)
7. Demkowicz, L., Gopalakrishnan, J., Niemi, A.H.: A class of discontinuous Petrov–Galerkin methods. Part III: Adaptivity. Appl. Numer. Math. (to appear)
8. Gopalakrishnan, J., Qiu, W.: An analysis of the practical DPG method. Math. Comp. **83**, 537–552 (2014)
9. Matsokin, A., Nepomnyaschikh, S.: A Schwarz alternating method in a subspace. Soviet Math. **29**, 78–84 (1985)
10. Monk, P.: Finite Element Methods for Maxwell's Equations. Oxford University Press, New York (2003)
11. Toselli, A., Widlund, O.: Domain Decomposition Methods - Algorithms and Theory. Springer, New York (2005)

A smooth transition approach between the Vlasov-Poisson and the Euler-Poisson system

Giacomo Dimarco[1], Luc Mieussens[2], and Vittorio Rispoli[3]

1 Introduction

Plasma dynamics is characterized by a wide range of spatial and temporal scales. Typical examples include plasmas produced around hypersonic bodies, ion wind of corona discharges, magnetic fusion processes. Depending on conditions, kinetic models of Boltzmann type or macroscopic models are commonly used for plasma physics simulations. The most common kinetic model for plasmas is the Vlasov equation, coupled with the electromagnetic field equations. On the other hand, Euler or Navier-Stokes based models coupled with the Maxwell equations are used for describing equilibrium plasma flows. Even if fluid models are sufficiently accurate to describe many observed phenomena, however, for some of them, this choice is inadequate. In these cases, it turns out that a kinetic description is strictly necessary to correctly represent the solutions. In these circumstances, the most widely used numerical methods for solving the Vlasov equation are Particle-In-Cell (PIC) approaches [1]. They have many advantages in terms of computational cost for large dimensional problems, for enforcing physical properties such as conservation laws and in terms of flexibility when handling with complex geometries. On the other hand, these methods involve a significant level of numerical noise and the convergence rate is in general quite slow. Moreover, in situations close to thermodynamical equilibrium, the cost of PIC methods or, more in general, direct Monte Carlo simulations increases. For this reason, domain decomposition techniques have been proposed in the recent past (see [2, 4, 5, 6, 8, 9]). Indeed, in many situations, the resolution of the kinetic equations in the whole computational domain is unnecessary because the fluid equations coupled with suitable equations for the electromagnetic fields provide a sufficiently accurate solution, except in small zones like shock layers or extremely rarefied regions where departure from thermodynamical equilibrium is strong.

In this paper, we focus on an adaptive kinetic-fluid approach which incorporates kinetic phenomena in selected regions of phase space where they play a fundamental role. More in detail, we propose a numerical method for the resolution of the collisional Vlasov-Poisson equation coupled with the compressible Euler-Poisson equations through a domain decomposition technique.

[1]Institut de Mathématiques de Toulouse; Université de Toulouse; UPS, INSA, UT1, UTM; CNRS, UMR 5219; F-31062 Toulouse, France. e-mail: giacomo.dimarco@math.univ-toulouse.fr .[2]Institut de Mathématiques de Bordeaux; Université de Bordeaux; 351, cours de la Libération - 33405 TALENCE cedex, France. e-mail: Luc.Mieussens@math.u-bordeaux1.fr .[3] Department of Mathematics; University of Ferrara; Via Machiavelli, 35 - 44121 Ferrara, Italy. e-mail: rspvtr@unife.it

The present paper represents an extension of two our earlier works [2, 4], in which we coupled the BGK equation and the compressible Euler equations. The key point on which the method relies is the introduction of a buffer zone in which the transition from the Vlasov-BGK-Poisson equations and the Euler-Poisson equations and vice-versa is gradual. Therefore, in the buffer zone, both models are solved and the solution of the full problem is obtained as the combination of the kinetic and fluid solutions. The introduction of the intermediate zone makes each of the models degenerate at the interfaces. In this way, no interface condition is needed. Finally, in this work we consider a constant in time coupling function and refer to [7] for the time dependent case.

2 The Vlasov-BGK-Poisson equation

We consider the collisional Vlasov equation for describing the ions evolution in a plasmas. In this work we assume that the electrons form a uniform neutralizing background. The binary interactions between particles are substituted by relaxation towards the equilibrium. The rescaled equation reads

$$\partial_t f + v \cdot \nabla_x f + \mathrm{E} \cdot \nabla_v f = \frac{1}{\tau}(M_f - f), \tag{1}$$

with the initial condition

$$f(x,v,t=0) = f_0(x,v), \tag{2}$$

where $f = f(x,v,t)$ is a non negative function describing the time evolution of the distribution of particles which move with velocity $v \in \mathbb{R}^d$ in the position $x \in \Omega \subset \mathbb{R}^d$ at time $t > 0$. In the general case, the relaxation time τ is a function of the macroscopic quantities. For our scopes, in the present paper, the relaxation frequency will be fixed and given at the beginning of the simulations. We refer to [7] for more physical cases. The electric field E is given as a gradient of a potential function $\mathrm{E} = \nabla_x \Phi$, where Φ is obtained from the solution of the Poisson equation

$$\lambda^2 \Delta \Phi = \int_{\mathbb{R}^d} f dv - \rho_0, \tag{3}$$

with λ the so called Debye length and ρ_0 the background electrons density. The local thermodynamical equilibrium is defined by

$$M_f = M_f[\rho,u,T](v) = \frac{\rho}{(2\pi\theta)^{d/2}} \exp\left(\frac{-|u-v|^2}{2\theta}\right), \tag{4}$$

where ρ and u are the density and mean velocity while $\theta = RT$ with T the temperature of the ions and R the gas constant.

Formally as $\varepsilon \to 0$ the function f tends to the local Maxwellian. In this limit, multiplying the Vlasov-BGK equation (1) by 1, v, $\frac{1}{2}|v^2|$ (the so-called collision in-

variants), and integrating with respect to v, leads to the following system of balance laws

$$\frac{\partial \rho}{\partial t} + \nabla_x \cdot (\rho u) = 0, \tag{5}$$

$$\frac{\partial \rho u}{\partial t} + \nabla_x \cdot (\rho u \otimes u + pI) - \rho \mathrm{E} = 0, \tag{6}$$

$$\frac{\partial \mathscr{E}}{\partial t} + \nabla_x \cdot ((\mathscr{E} + p)u) - \rho u E = 0, \tag{7}$$

$$p = \rho\theta, \quad \mathscr{E} = \frac{d}{2}\rho\theta + \frac{1}{2}\rho|u|^2. \tag{8}$$

where p is the pressure and $\mathscr{E}$ the total energy.

3 The coupling method

In this section we present the coupling strategy between the Vlasov-BGK-Poisson equations and the Euler-Poisson system. We will deal here with a constant in time coupling between the micro and macroscopic models. However, our final scope is to derive a time dependent coupling strategy, we refer to [7] for this case. We also refer to [7] for more details on the numerical discretization, the treatment of the boundary conditions and for the general theory about the time-dependent case.

3.1 Decomposition of the kinetic equation

The coupling strategy is inspired by two recent works ([2, 4]) in which the rarefied gas dynamic case was considered. For sake of simplicity we describe the method in one space and velocity dimensions. It can be easily extended to a generic N-dimensional setting. Also different meshes for the cut-off function and for the other variables can be used.

We denote the buffer interval by $[a,b]$, and we introduce a cut-off function $h(x)$ such that

$$h(x) = \begin{cases} 1, & \text{for } x \le a \\ 0, & \text{for } x \ge b \\ 0 \le h(x) \le 1, & \text{for } x \in [a,b] \end{cases} \tag{9}$$

For instance, h can be chosen piecewise linear in $[a,b]$:

$$h(x) = \frac{x-b}{a-b} \quad \text{for} \quad x \in [a,b].$$

We define two distribution functions such that $f_R = hf$ while $f_L = (1-h)f$. We look now for an evolution equation for f_R and for f_L. We write

$$\partial_t f_R = \partial_t(hf) = h\partial_t f,$$
$$\partial_t f_L = \partial_t((1-h)f) = (1-h)\partial_t f.$$

Thus multiplying the Vlasov-BGK equation (1) by h and $1-h$ respectively, we obtain the following equations for the time evolution of the distributions f_R and f_L

$$\partial_t f_R = h\Big(-v\partial_x f - \mathrm{E}\cdot\nabla_v f + \frac{1}{\tau}(M_f - f)\Big),$$
$$\partial_t f_L = (1-h)\Big(-v\partial_x f - \mathrm{E}\cdot\nabla_v f + \frac{1}{\tau}(M_f - f)\Big),$$

which finally leads to the following system

$$\partial_t f_R + hv\partial_x f_R + hv\partial_x f_L + \mathrm{E}\partial_v f_R = \frac{h}{\tau}(M_f - f), \tag{10}$$
$$\partial_t f_L + (1-h)v\partial_x f_L + (1-h)v\partial_x f_R + \mathrm{E}\partial_v f_L = \frac{1-h}{\tau}(M_f - f), \tag{11}$$
$$f = f_R + f_L, \tag{12}$$

with initial data

$$f_R(x,v,0) = h(x,0)\,f(x,v,0)\,, \quad f_L(x,v,0) = (1-h(x,0))\,f(x,v,0). \tag{13}$$

It is important to note that if $f = f_L + f_R$ is the solution of (1) with initial data (2), then (f_L, f_R) is the solution of (10-11) with initial data (13) and conversely.

3.2 Kinetic-Hydrodynamic coupling

Now, let us assume that the domain can be subdivided into two regions: in one of the regions, the distribution function is close to a local Maxwellian while in the other, it is far from it. We choose to set $h = 0$ in the region where f is close to the Maxwellian. Therefore, f_L is close to its associated Maxwellian M_{f_L} and we can replace the Vlasov-BGK equation (1) by its macroscopic limit equations without making any significant error. We also suppose that in the buffer zone, f_L remains close to the equilibrium and thus, it can be replaced by M_{f_L} in the whole interval $x < b$.

Replacing f_L by M_{f_L} in (11) and taking the hydrodynamic moments (mass, momentum and energy), leads to the following modified Euler system defined in the interval $x \leq b$

$$\begin{aligned}
&\frac{\partial \rho_L}{\partial t} + (1-h)\partial_x(\rho_L u_L) = -(1-h)\partial_x\left(\int_{\mathbb{R}} v f_R\, dv\right),\\
&\frac{\partial \rho_L u_L}{\partial t} + (1-h)\partial_x(\rho_L u_L^2 + p_L) - \mathrm{E}\rho_L = -(1-h)\partial_x\left(\int_{\mathbb{R}} v^2 f_R\, dv\right), \qquad (14)\\
&\frac{\partial \mathscr{E}_L}{\partial t} + (1-h)\partial_x((\mathscr{E}_L + p_L)u_L) - \rho_L u_L \mathrm{E} = -(1-h)\partial_x\left(\int_{\mathbb{R}} v\frac{|v|^2}{2} f_R\, dv\right),
\end{aligned}$$

with initial data

$$(\rho_L, u_L, \theta_L)|_{(x,0)} = (1-h|_{(x,0)})(\rho, u, \theta)|_{(x,0)}.$$

Under these assumptions, we have $f = f_R + M_{f_L}$, where f_R is a solution of:

$$\partial_t f_R + hv\partial_x f_R + hv\partial_x M_{f_L} + \mathrm{E}\partial_v f_R = \frac{h}{\tau}(M_f - f), \qquad (15)$$

in the interval $x \geq a$. Thus, the coupling model consists of system (14) for the hydrodynamic moments in the region $x \leq b$ and of equation (15) for the kinetic distribution function in the region $x \geq a$.

When $h = 0$, system (14) coincides with system (8) because $f_R = 0$ and $f_L = M_{f_L}$. Moreover no boundary conditions are needed at the boundary $x = b$ because the spatial derivatives are degenerate at $x = b$ for the fluid model. A similar remark is true for f_R. Indeed, when $h = 0$, $f_R = 0$ and no boundary conditions are needed for the kinetic equation at $x = a$ because the spatial derivatives are degenerate in equation (15). In the buffer zone $[a,b]$, the solution of the full kinetic problem f is computed as the sum of the Maxwellian M_{f_L} and of the function f_R. To summarize, the solution of the full kinetic problem is given by f_R if $x > b$, by M_{f_L} if $x < a$ and by $M_{f_L} + f_R$ if $x \in [a,b]$.

An important feature of the method is that it is very easy to divide the domain in more than two zones. Thus we can define as many buffers and as many kinetic regions as necessary if the macroscopic model fails to give the correct solution in different parts of the domain which are far apart from each other. In this latter case, the function h is still a piecewise linear function but there are multiple buffer zones $[a_j, b_j]$. Additionally, we can create new buffer zones and new kinetic zones during the simulation. Such strategy is presented in [4] for the Boltzmann-BGK and in [7] for the Vlasov-Poisson equations.

4 Numerical test

The numerical example we present is a one-dimensional plasma expansion problem. This is a two-species problem composed by free ions and fixed electrons. Ions initially occupy a small region of thickness D of the space where they have an high density while in the rest of the domain they have very small density. Background electrons are initialized by a Maxwell-Boltzmann equilibrium with a self-consistent

potential and their density is constant everywhere. The test problem consists in observing the expansion of the ions.

This kind of phenomena are well described by the Vlasov-Poisson system in rarefied regions and by the Euler-Poisson system in dense regions. For this test case, we consider all our equation in their adimensional form. for dimensional test cases we refer to [7].

The numerical physical domain goes from the left boundary $x_L = -20$ to the right boundary at $x_R = 20$ while the velocity domain goes from $v_{\min} = -200$ to $v_{\max} = 200$. There are 1000 cells in physical space and 140 cells in velocity space. The slab where ions are initialized with high density is $[x_L, D]$ with $D = -8$. Initial conditions are as follows: for ions, the density is $\rho = 1$, mean velocity $u = 0$ and temperature $T = 10$ in the high-density slab $[x_L, D]$ while in the remaining part of the domain the density is $\rho = 5 \times 10^{-2}$, mean velocity $u = 0$ and temperature $T = 8$. Electrons are initialized with density $\rho_0 = 1$ everywhere.

The collision frequency is given by $1/\tau$ where $\tau = 5\ 10^{-6}$ in the hydrodynamic part and $\tau = 10^{-1}$ in the kinetic part. The Debye length takes the value $\lambda^2 = 10^{-2}$ and Dirichlet boundary conditions are imposed for the electric potential as $\Phi(x_L) = 0$ and $\Phi(x_R) = 10.0$.

The cut-off function h is initialized as $h = 0$ for x ranging from -20 to $a = -1.0$ (fluid region), $h = \frac{x-a}{b-a}$ with $b = 0.5$ (buffer zone) and $h = 1$ for $x > b$ (kinetic region).

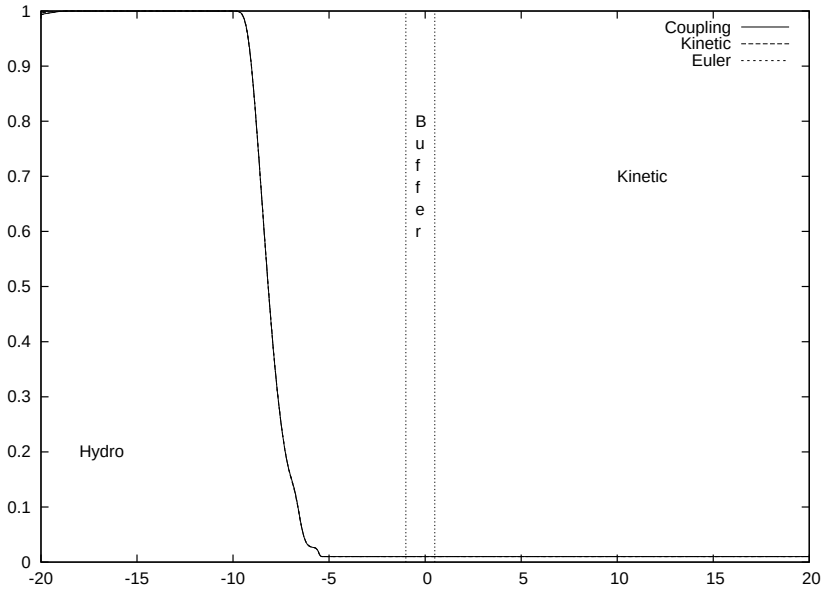


Fig. 1 Density profile for the ions at time $t = 3.6 \times 10^{-3}$. Continuous line coupling method, dashed line Vlasov-BGK-Poisson model, dotted line Euler-Poisson system.

Boundary conditions are treated as a constant incoming maxwellian injection at the left boundary and as free Neumann conditions at the right one.

When the simulation begins, ions start to expand. We plot the solution for the density after few time steps $t_i = 3.6 \times 10^{-3}$ s (Fig. 1), at an intermediate state before reaching the buffer zone at $t_m = 8.4 \times 10^{-3}$ s (Fig. 2) and at the end of the simulation at $t_f = 1.32 \times 10^{-2}$ s (Fig. 3). In the figures we report the results of the domain decomposition strategy together with the results obtained by employing a scheme which solves the Vlasov-BGK-Poisson equation everywhere. We also report the results obtained by solving the Euler-Poisson system in all the domain.

The figures show that during all the simulation the coupling strategy is able to capture the good solution which is the one furnished by solving the kinetic equation everywhere, while the scheme which solves the Euler-Poisson system fails in describing the good solution where the collision frequency is very small. For more realistic problems, it will be necessary to follow the discontinuities and the regions where the rarefaction is high during the time evolution of the problem. This can be accomplished by constructing some adaptive and dynamic decomposition which are the subject of a future work [7].

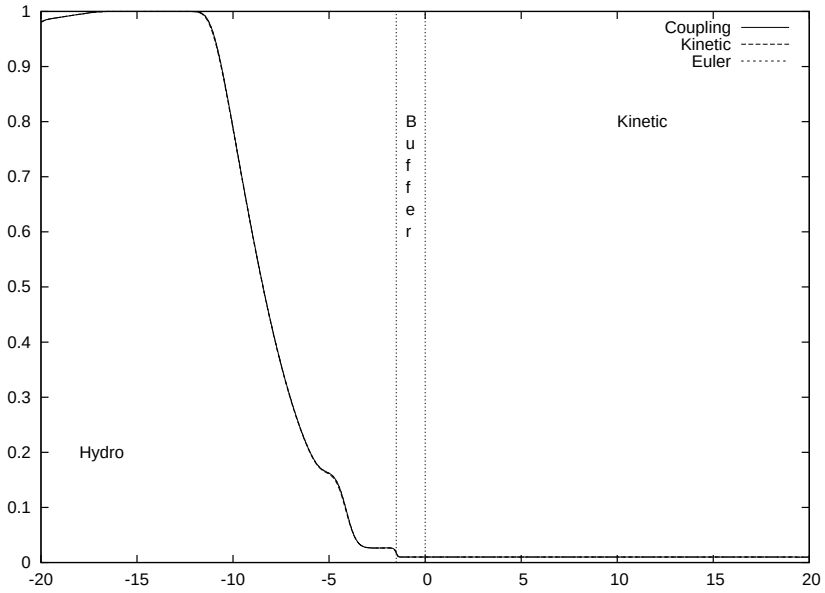


Fig. 2 Density profile for the ions at time $t = 8.4 \times 10^{-3}$. Continuous line coupling method, dashed line Vlasov-BGK-Poisson model, dotted line Euler-Poisson system.

References

1. C.K. BIRDSALL, A.B. LANGDON, *Plasma Physics via Computer Simulations*, Taylor and Fransis, (2004).
2. P. DEGOND, S. JIN, L. MIEUSSENS, *A Smooth Transition Between Kinetic and Hydrodynamic Equations*, J. Comput. Phys., Vol. 209, pp. 665–694 (2005).
3. P. DEGOND, F. DELUZET, L. NAVORET, A.B. SUN, Asymptotic-Preserving Particle-In-Cell method for the Vlasov-Poisson system near quasineutrality J. Comput. Phys., Vol. 229, pp. 5630–5652 (2010).
4. P. DEGOND, G. DIMARCO, L. MIEUSSENS, *A moving interface method for dynamic kinetic-fluid coupling*, J. Comput. Phys., Vol. 227, pp. 1176–1208 (2007).
5. P. DEGOND, G. DIMARCO, L. MIEUSSENS, *A multiscale kinetic-fluid solver with dynamic localization of kinetic effects*, J. Comput. Phys., Vol. 229, pp. 4907–4933 (2010).
6. P. DEGOND, G. DIMARCO, *Fluid simulations with localized Boltzmann upscaling by direct Monte Carlo*, J. Comput. Phys., Vol. 231, pp. 2414–2437 (2012).
7. G. DIMARCO, L. MIEUSSENS, V. RISPOLI, *A moving interface method for connecting the Vlasov-Poisson system to the Euler-Poisson system*, in preparation.
8. V.I. KOLOBOV, R.R. ARSLANBEKOV,V.V. ARISTOV, A.A. FROLOVA, S.A. ZABELOK, *Towards adaptive kinetic-fluid simulations of weakly ionized plasmas*, J. Comput. Phys., Vol 231, pp. 839-869 (2012).
9. S. TIWARI, *Coupling of the Boltzmann and Euler equations with automatic domain decomposition*, J. Comput. Phys., Vol. 144, pp. 710–726 (1998).

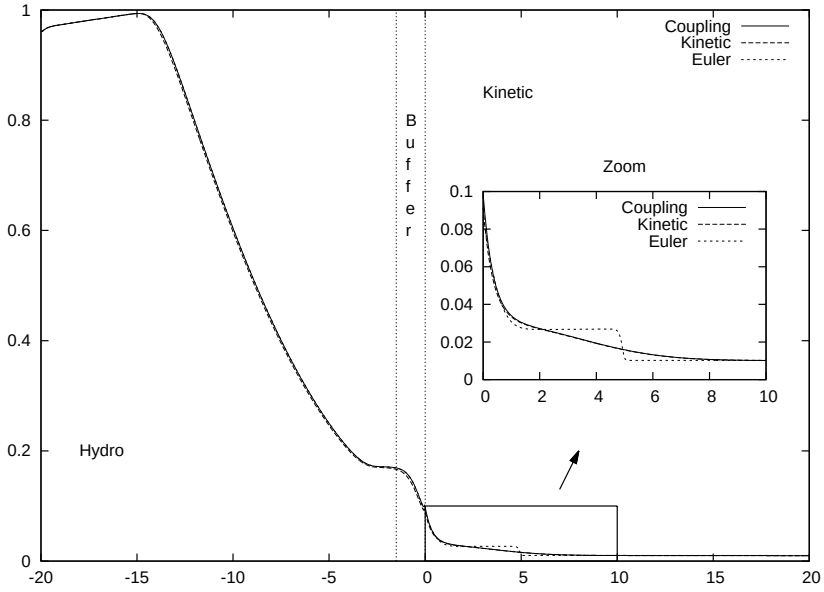


Fig. 3 Density profile for the ions at time $t = 1.32 \times 10^{-2}$. Continuous line coupling method, dashed line Vlasov-BGK-Poisson model, dotted line Euler-Poisson system.

The parareal in time algorithm applied to the kinetic neutron diffusion equation

A.-M. Baudron[1,3], J.-J. Lautard[1,3], Y. Maday[2,3,4,5], and O. Mula[1,2,3]

Introduction

In the framework of nuclear core calculations, the development of efficient tools to run neutron kinetic computations is a field of current active research. While such calculations are crucial for security assessment and the study of new reactor concepts, they present several mathematical and computational issues that still need to be overcome.

The exact model (kinetic transport equation) is indeed far too expensive to be simulated for these purposes and different simplifications (multi group diffusion approximation) have led to more tractable numerical simulations. Nevertheless, on real geometries and despite the use of domain decomposition enabling accelerations of the simulations thanks to parallel architectures [7], there is still need for improvements for applications on regular basis.

In this context, the purpose of this work is to investigate the implementation of the parareal in time algorithm [9] within an industrial solver called MINOS developed at C.E.A. (cf. [4]) following the preliminary analysis [5].

The paper is organized as follows: after the presentation of the neutron diffusion equation in Section 1, the main aspects of the parareal method will be recalled in Section 2. In particular, we will explain the distributed algorithm that has been used in our case from the point of view of the expected speed-up. The performances of the parareal in time algorithm in a numerical application are summarized in section 3 which is followed in Section 4 by a discussion about the convergence behavior observed in our example.

1 Model

The evolution of the flux ψ of neutrons in a reactor core $\mathscr{R}$ is governed by a kinetic transport PDE whose theoretical properties (existence, uniqueness, positiveness of the solution) have been investigated in e.g. [6] (chapter XXI, section 2, theorem 3). Given the fact that ψ depends on 7 variables, namely the time t, the position within

[1] C.E.A, CEA Saclay - DEN/DANS/DM2S/SERMA/LLPR - 91191 Gif-Sur-Yvette CEDEX - France , e-mail: {anne-marie.baudron}{jean-jacques.lautard}{olga.mulahernandez}@cea.fr .[2] UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France e-mail: maday@ann.jussieu.fr .[3] LRC MANON, Laboratoire de Recherche Conventionnée, CEA/DEN/DANS/DM2S and UPMC-CNRS/LJLL. .[4] Institut Universitaire de France .[5] Brown Univ, Division of Applied Maths, Providence, RI, USA

the reactor denoted as $\overrightarrow{r}$, the velocity of the neutrons $\overrightarrow{v} = \sqrt{2E/m}\overrightarrow{\Omega}$ where E stands for the energy of the neutron, $\overrightarrow{\Omega}$ stands for the direction of the velocity and m is the mass of the neutron, it has been proposed in e.g. [6] (chapter XXI, section 5), to simplify the model by first considering the average flux over the angular variables as the unknown: $\phi(t,\overrightarrow{r},E) = \frac{1}{4\pi}\int_{\mathbb{S}_2}\psi(t,\overrightarrow{r},\overrightarrow{\Omega'},E)d\overrightarrow{\Omega'}$. This approach leads to results that are accurate enough in most of the usual cases but the computing time still remains unacceptably long.

Another simplification consists in averaging also in the energy variable. This further approximation, known as the multi-group theory [10], is based on the division of the energy interval into G subintervals ($[E_{min},E_{max}] = [E_G,E_{G-1}]\cup\cdots\cup[E_1,E_0]$) and leads to consider the set $\boldsymbol{\Phi} = \{\phi^g\}_{g\in\{1,G\}}$ as the new unknown solution. In order to take into account the presence of radioactive isotopes (also called precursors) that are important since they emit neutrons with a given delay, the model is complemented with a set of first order ODE's expressing their decays denoted as $\mathbf{C} = \{C_\ell\}_{\ell\in\{1,L\}}$. Since their half-lives have values that vary in a wide range, the resulting system is very stiff and small time steps are required for an accurate approximation in long time intervals.

The set $(\boldsymbol{\Phi},\mathbf{C})$ is the solution of the following set of multi-group diffusion equations:

$$(*)\begin{cases} \frac{1}{v^g}\frac{\partial\phi^g}{\partial t} - \nabla.\left(D^g\overrightarrow{\nabla}\phi^g\right) + \sigma_t^g\phi^g = \sum\limits_{g'=1}^{G}\mathscr{S}^{gg'}\phi^{g'} + \chi_p^g\sum\limits_{g'=1}^{G}\mathscr{F}^{g'}\phi^{g'} + \sum\limits_{\ell=1}^{L}\chi_\ell^g\lambda_\ell C_\ell \\ \text{over } [0,T]\times\mathscr{R},\forall g\in\{1,G\}, \\ \frac{\partial C_\ell}{\partial t} = -\lambda_\ell C_\ell + \sum\limits_{g'=1}^{G}\mathscr{F}_\ell^{g'}\phi^{g'} \text{ over } [0,T]\times\mathscr{R},\forall\ell\in\{1,L\}, \\ \phi^g = 0, \text{ on } [0,T]\times\partial\mathscr{R} \\ \phi^g(0,.) = \phi_0^g(.);\ C_\ell(0,.) = C_{\ell,0}(.) \text{ on } \mathscr{R} \end{cases}$$

where v^g is the neutron velocity, D^g the diffusion coefficient and σ_t^g the total cross-section in energy group g. χ_p^g is the prompt spectrum in energy group g, χ_ℓ^g the delayed spectrum of precursor ℓ in energy group g and λ_ℓ is the decay constant of precursor ℓ. $\mathscr{F}^g$ and $\mathscr{F}_\ell^g$ denote the prompt and delayed fission operators respectively. $\mathscr{S}^{gg'}$ is the neutron scattering operator from energy g to g' and makes the flux equations be coupled with respect to the energy variable.

2 The parareal algorithm

The unsteady problem $(*)$ can be written in a more compact form:

$$\frac{\partial y}{\partial t} + \mathscr{A}(t;y) = 0\ , t\in[\tau_0,\tau_1]; \tag{1}$$

it is complemented with initial conditions at time $t = \tau_0 : y(\tau_0) = y_0$. The parareal in time algorithm applied to (1) is an iterative technique where, at each iteration a predictor corrector propagation is proposed based on two propagators : a fine one $\mathscr{F}_{\tau_0}^{\tau_1}(y_0)$ that computes an approximation of the solution of (1) at time τ_1 accurately but slowly, and a coarse one $\mathscr{G}_{\tau_0}^{\tau_1}(y_0)$ that computes an other approximation quickly but not so accurately (and not accurately enough). In addition to these two propagators $\mathscr{F}$ and $\mathscr{G}$, the parareal in time algorithm is based on the division of the full interval $[0,T]$ into N sub-intervals $[0,T] = \bigcup_{n=0}^{N-1}[T_n, T_{n+1}]$ that will each be assigned to a processor P_n, assuming that we have N processors at our disposal.

The value $y(T_n)$ is approximated by Y_n^k as k increases with an accuracy that tends to the one achieved by the fine solver (see [9], [2], [3] for further details). It is obtained by the recurrence relation:

$$Y_{n+1}^{k+1} = \mathscr{G}_{T_n}^{T_{n+1}}(Y_n^{k+1}) + \mathscr{F}_{T_n}^{T_{n+1}}(Y_n^k) - \mathscr{G}_{T_n}^{T_{n+1}}(Y_n^k),\ n = 1,...,N \tag{2}$$

starting from $Y_{n+1}^0 = \mathscr{G}_{T_n}^{T_{n+1}}(Y_n^0)$ In this work, the recently described distributed algorithm (summarized in [1]) has been used for the practical implementation of parareal. It represents an improvement of parareal from the algorithmic point of vue.

The first method of implementation was indeed suggested in [9] and consisted on a master-slave algorithm where the master carried out the coarse propagation in the whole time interval (each slave being in charge of the fine propagations over its assigned time slice and sending $\mathscr{F}_{T_n}^{T_{n+1}}(Y_n^k)$ to the master so that the master computed the parareal corrections (2) $\forall n$). This original algorithm gives rise to two main computing drawbacks: the coarse propagation by the master is a bottleneck in the computation and the memory requirement in the master processor scales linearly with the number of slaves. The distributed algorithm improves both aspects and can easily be implemented via the MPI library: for each processor P_n the fine and the coarse solvers are propagated over $[T_n, T_{n+1}]$ and the parareal correction Y_{n+1}^{k+1} is carried out. The process is repeated until convergence, i.e. $\|Y_n^{k+1} - Y_n^k\| < \eta,\ \forall n$, where η is a given tolerance.

It is easy to realize that this kind of implementation does not change the number of iterations in order the parareal algorithm to converge but it provides better speed-ups than the original master-slave version. This is the reason why the distributed algorithm has been implemented in this study. Indeed, if we do not take into account the communication time between processors, the theoretical speed-ups of the distributed and master-slave algorithms are respectively (see [1]):

$$S_{distrib} = \frac{N}{Nr + k^*(1+r)} \quad ; \quad S_{MS} = \frac{N}{Nr(1+k^*) + k^*} \tag{3}$$

where r is the ratio between the two solution times of the two propagators $\mathscr{G}$ and $\mathscr{F}$ and k^* is the number of parareal iterations needed in order to converge.

3 Numerical simulation

3.1 Definition of the test case:

The parareal algorithm has been implemented with an implicit discretization in time. Note that here we have used the same physical model (diffusion) for both the coarse and the fine solvers (the only difference is the size of the time steps used to solve equation $(*)$ δt for $\mathscr{F}$ and $\Delta t = T_{n+1} - T_n$ for $\mathscr{G}$). At each time step, a Gauss-Seidel iteration is used on the energy groups and the spatial discretization is done with RT-1 finite elements (see [4]).

The geometry and history that have been chosen for the simulation is the so called TWIGL benchmark that represents a rod withdrawal (see [8]). The geometry of the core is three-dimensional. A cross-sectional view of it is specified in FIGURE 1 where only a quarter of it has been represented (the rest can be inferred by symmetry). The first group of rods (yellow) is withdrawn from $t = 0$ ($z = 100$ *cm* measured starting from below) until $t = 26.6$ *s*. ($z = 180$ *cm*) at a constant velocity. The second group of rods (brawn) is inserted from $t = 7.5$ *s*. ($z = 180$ *cm*) until $t = 47.7$ *s*. ($z = 60$ *cm*) and the simulated interval of time is $[0,T]$ with $T = 66.6$ *s*.

Computations have been carried out with $G = 2$ energy groups, $L = 6$ precursors. The coefficients of $(*)$ remain constant in time and only the geometry varies. The fine solver has a fixed time step of $\delta t = 1/6$ *s*.

The scaling has been evaluated with a convergence test associated in which the tolerance η has been fixed to the precision of the numerical scheme (i. e. $\eta \sim 10^{-3}$). With this threshold, convergence has been achieved after only $k^* = 2$, 3 or at most 4 iterations of the parareal in time algorithm.

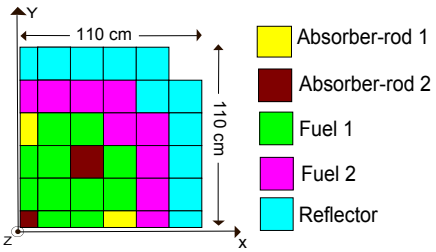


Fig. 1 Cross-sectional view of a quarter of the core in the TWIGL benchmark

3.2 Strong scaling results:

For the strong scaling analysis, the same problem has been solved on an increasing number N of processors. The size of each interval, equal to the time step of the coarse solver, has been reduced from $\Delta t = 50\delta t$ to $\Delta t = 5\delta t$ in order to increase the number of processors. Therefore, as N varies, the ratio r and the number of parareal iterations k^* change. With the computed k^* and using $\delta t/\Delta t$ as an approximation of r, one can infer from formula 3 the optimal speed-up values that can be obtained in our current case with the distributed algorithm (measured speed-ups are of course lower due to the communication time that is not taken into account in formula 3). The values are plotted in FIGURE 2, where the theoretical speed-ups of the master-slave algorithm are also shown in order to compare both methods.

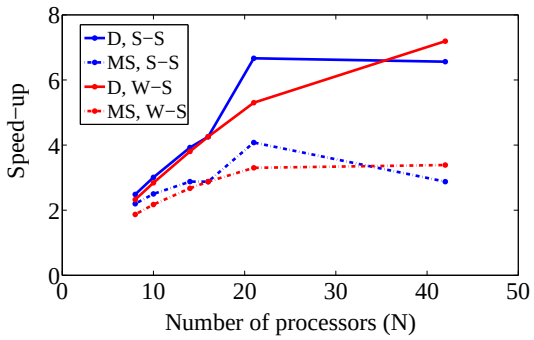


Fig. 2 Optimal speed-ups obtained for the scaling tests (D=Distributed algorithm; MS= Master-Slave algorithm; S-S= Strong Scaling; W-S= Weak Scaling)

As it can be observed, the distributed algorithm performs better for any number N of processors. For a reduced number of processors, the speed-ups are similar because both algorithms increase like N/k^* for N small enough. However, when N becomes significant in formulae 3, the distributed algorithm will behave like N/r and the master-slave method like $N/(r(1+k^*))$, making the distributed algorithm become more perfomant on a wider range of values of N. The performances reach a plateau and even decrease when N becomes very large ($N > 20$ in our case) because the cost of $\mathscr{G}$ becomes equivalent to the cost of $\mathscr{F}$ (r tends to 1).

3.3 Weak scaling results

For this alternative evaluation of the scaling, the same geometry as before has been used. We now consider the case in which the problem has a variable length $T = N\Delta t$

and the time step of the coarse solver Δt is fixed (i.e. the size of the problem linearly increases with the number N of processors). For our computations, the fine and coarse time steps are fixed to $\delta t = 1/6\ s.$ and $\Delta t = 50\delta t$ respectively.

The control rods are inserted and withdrawn periodically with a sequence of motion that creates fluctuations in the total power. With the computed k^*, the optimal speed-ups for the distributed algorithm are plotted in FIGURE 2 and compared to the master-slave model. The most important result here is that the distributed algorithm can effectively speed-up long time calculations as it can be observed. When compared to the master-slave implementation for large values of N, the distributed algorithm has a clear advantage because the increase of k^* has not such a strong negative impact on it than on the master-slave implementation (as it can also be seen in FIGURE 2).

4 About the convergence of parareal in the kinetic neutron diffusion equation

The analysis of the convergence process can be done into two ways, either by looking only at the history of the values at each T_n, $1 \leq n \leq N$, or by looking at the error at each fine discrete time $m\Delta t$:

$$e^k(t_n + m\delta t)_{fine} = \frac{\|\mathscr{F}_{T_n}^{T_n+m\delta t}(\Phi_n^k) - \mathscr{F}_0^{T_n+m\delta t}(\Phi_0)\|_{L^2}}{\|\Phi_0\|_{L^2}} \tag{4}$$
$$\forall n = 1,...N,\ \forall m = 0,1,...\frac{\Delta t}{\delta t},\ \forall k = 0,...,N-1$$

FIGURE 3 illustrates the global convergence history according to formula 4. Above the convergence threshold, we note a surprising behavior of the error over each interval $[T_n, T_{n+1}]$ that is, in most cases, neither linear nor constant despite that $(*)$ is linear. The following analysis will explain that this is due to the presence of the radioactive isotopes.

Under several hypothesis (see the point kinetics approximation in [10]), the kinetic behavior of system $(*)$ can be analysed through a set of first order ODE's of the form:

$$(5)\ \begin{cases} \frac{d\Phi(t)}{dt} = \alpha\Phi(t) + \sum_{\ell=1}^{L} \lambda_\ell C_\ell(t) \\ \frac{dC_\ell(t)}{dt} = \gamma_\ell \Phi(t) - \lambda_\ell C_\ell(t), \forall \ell = 1,\ldots,L \\ \Phi(0) = \Phi_0,\ C_\ell(0) = C_{\ell,0} \end{cases}$$

where the coefficients are in the range $-0.5 \leq \alpha \leq -6.10^{-3}$, while for any $\ell, 1 \leq \ell \leq L$, $10^{-2} \leq \lambda_\ell \leq 4$ and $3.10^{-3} \leq \gamma_\ell \leq 3,4.10^{-2}$

In order to understand the phenomenon in the simulation of $(*)$ represented in FIGURE 3, let us consider the case where $L = 1$ in (5). Due to linearity, the evolution of the error (e_{fine}) between the parareal fine propagator and the sequential fine one follows the same evolution as Φ in (5) over each interval $[T_n, T_{n+1}]$ starting from

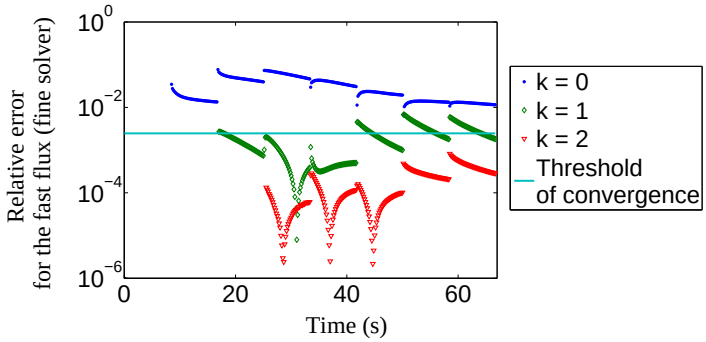


Fig. 3 Example of convergence of the fine parareal solution $\mathscr{F}_{T_n}^{T_n+m\delta t}(\Phi_n^k)$ (TWIGL benchmark, $N = 8$ processors, $\Delta t = 8.3\ s$.)

an initial error $\delta\Phi$ over Φ and δC over $C = C_1$. This system can be solved and the solution is the sum of two exponential behaviors $e^{\mu_- t}$ and $e^{\mu_+ t}$ where $\mu_\pm$ are the two eigenvalues associated with the problem : $\mu_\pm = \frac{(\alpha-\lambda)\pm\sqrt{(\lambda+\alpha)^2+4\lambda\gamma}}{2}$. In the range of values where the physical parameters lie, $\lambda+\alpha$ is not small and we can consider that $\gamma = \frac{(\lambda+\alpha)^2}{4\lambda}(\varepsilon+\circ(\varepsilon))$. In this case, the eigenvalues behave as $\mu_\pm = \frac{\alpha-\lambda\pm|\lambda+\alpha|}{2} \pm \frac{|\lambda+\alpha|}{4}\varepsilon+\circ(\varepsilon)$ where ε is a small quantity, the error $\delta\Phi(t) = \delta\Phi_0 e^{\alpha t} + \frac{\lambda}{\lambda+\alpha}\delta C_0\left(e^{\alpha t}-e^{-\lambda t}\right) + \theta(\delta\Phi_0,\delta C_0,\alpha,\lambda)\varepsilon+\circ(\varepsilon)$, with θ gathering the terms at order ε. At first order, and depending on the values of α and λ, $\delta\Phi$ (and therefore e_{fine}) will present an exponentially decreasing trend (e.g. $\alpha = -0.006$, $\lambda = 4$) or a brief increase followed by a decrease (e.g. $\alpha = -0.5$, $\lambda = 0.01$) as it appears in FIGURE 3.

Conclusion

The results of this study show that the parareal distributed algorithm can effectively speed-up neutron kinetic diffusion calculations. They can certainly be improved by coupling parareal with spatial domain decomposition. A further analysis needs to be done on the impact of the communication time between processors.

An analysis of a surprising behavior of the error within each interval $[T_n, T_{n+1}]$ has also been explained and is a consequence of a special tune of the parameters.

Note also that these results represent the first implementation of the parareal in time algorithm within the industrial solver MINOS so the current results represent as well a successful industrial application of parareal.

These results are encouraging because they open the door to the construction of kinetic transport solvers. Our ongoing study is therefore to explore whether the parareal algorithm can successfully accelerate such calculations.

Acknowledgements This work was supported in part by the joint research program MANON between CEA-Saclay and University Pierre et Marie Curie-Paris 6.

References

1. Aubanel, E.: Scheduling of tasks in the parareal algorithm. Parallel Computing **37**, 172–182 (2011)
2. Baffico, L., Bernard, S., Maday, Y., Turinici, G., Zérah, G.: Parallel-in-time molecular-dynamics simulations. Phys. Rev. E **66** (2002)
3. Bal, G., Maday, Y.: A parareal time discretization for non-linear PDE's with application to the pricing of an American put. Recent developments in domain decomposition methods **23**, 189–202 (2002)
4. Baudron, A.M., Lautard, J.J.: A simplified P_n solver for core calculation. Nuclear Science and Engineering **155**, 250–263 (2007)
5. Baudron, A.M., Lautard, J.J., Riahi, K., Maday, Y., Salomon, J.: Time-parareal parallel in time integrator solver for time-dependent neutron diffusion equation. Submitted
6. Dautray, R., Lions, J.L.: Analyse mathématique et calcul numérique. Masson, Cambridge, Massachusetts (1984)
7. Jamelot, E., Baudron, A.M., Lautard, J.J.: Domain decomposition for the SPN solver MINOS. Transport Theory and Statistical Physics **41**:7, 495–512 (2012)
8. Langenbuch, S., Maurer, W., Werner, W.: Coarse-mesh flux expansion method for the analysis of space-time effects in large light water reactor cores. Nuclear Science and Engineering **63**, 437–456 (1977)
9. Lions, J., Maday, Y., Turinici, G.: Résolution d'EDP par un schéma en temps pararéel. C. R. Acad. Sci. Paris (2001). T. 332, Série I, p. 661-668
10. Reuss, P.: Précis de neutronique. EDP Sciences, Collection Génie Atomique (2003)

Achieving robustness through coarse space enrichment in the two level Schwarz framework.

Nicole Spillane[1,2], Victorita Dolean[3], Patrice Hauret[2], Frédéric Nataf[1], Clemens Pechstein[4], and Robert Scheichl[5]

As many domain decomposition methods the two level Additive Schwarz method may suffer from a lack of robustness with respect to coefficient variation in the underlying set of PDEs. This is the case in particular if the partition into subdomains is not aligned with all jumps in the coefficients. Thanks to the theoretical analysis of two level Schwarz methods (see [11] and references therein) this lack of robustness can be traced back to the so called stable splitting property (already in [4]). Following the same ideas as in the pioneering work [1] we propose to solve a generalized eigenvalue problem in each subdomain which identifies which vectors are responsible for slow convergence. The spectral problem is specifically chosen to separate components that violate the stable splitting property. These vectors are then used to span the coarse space which is taken care of by a direct solve while all remaining components can be resolved on the subdomains. The result is a preconditioned system with a condition number estimate that does not depend on the number of subdomains or any jumps in the coefficients. We refer to this method as GenEO for Generalized Eigenproblems in the Overlaps. It is closely related to the work of [2] where the same strategy leads to a different eigenproblem and different condition number estimate (which also does not depend on the jumps in the coefficients or on the number of subdomains). A full theoretical analysis of the two level Additive Schwarz method with the GenEO coarse space (first briefly introduced in [8]) is given in [7]. Here our purpose is to show the steps leading from the abstract Schwarz theory to the choice of our generalized eigenvalue problem (5). In the first section we introduce the rather wide range of problems to which the method applies and give the classical two-level Schwarz condition number estimate in the abstract framework (again, see [11] and references therein). In the second section we work to make this condition local (on each subdomain), identify the GenEO generalized eigenproblem and state our main result (Theorem 2). Finally in the third section we illustrate the result numerically.

[1]Laboratoire Jacques-Louis Lions, CNRS UMR 7598, Université Pierre et Marie Curie, 75005 Paris, France .[2]Manufacture des Pneumatiques Michelin, 63040 Clermont-Ferrand, Cedex 09, France. .[3]Laboratoire J.-A. Dieudonné, CNRS UMR 6621, Université de Nice-Sophia Antipolis, 06108 Nice Cedex 02, France. .[4]Institute of Computational Mathematics, Johannes Kepler University, Altenberger Str. 69, 4040 Linz, Austria. .[5] Department of Mathematical Sciences, University of Bath, Bath BA27AY, UK.

1 Problem Setting

Given a finite dimensional Hilbert space V_h, a continuous and coercive bilinear form $a : V_h \times V_h \to \mathbb{R}$ and a right hand side $f \in V_h'$ we consider the following problem. Find $v \in V_h$ such that $a(v,w) = \langle f, w\rangle$ for all $w \in V_h$. Then given a basis for V_h we can derive a linear system $Av = f$.

Assumption: The following assumption is needed on the bilinear form: a is given through positive semi definite element matrices $\{a_\tau\}_{\tau\in\mathscr{T}_h}$ where $\mathscr{T}_h$ is a mesh on the computational domain Ω underlying V_h. Our method can also be defined for abstract elements and degrees of freedom as in [7] but here we focus on PDEs and prefer this more intuitive point of view.

The reason why we require this assumption is so that we may define, for any subset D which is resolved by the mesh, the following local bilinear form:

$$a_D(v,w) := \sum_{\tau\subset D} a_\tau(v_{|\tau}, w_{|\tau}). \tag{1}$$

The Additive Schwarz method is based on an overlapping partition $\{\Omega_j\}_{j=1}^N$ of Ω where each Ω_j is resolved by the mesh. On each of these subdomains, we denote the space of functions supported in Ω_j by: $V_{h,0}(\Omega_j) := \{v_{|\Omega_j} : v \in V_h,\ \mathrm{supp}(v) \subset \Omega_j\}$.

An important role is played by the extension operator $R_j^\top : V_{h,0}(\Omega_j) \to V_h$ which returns the extension by zero of a function $v \in V_{h,0}(\Omega_j)$ to Ω. The adjoint of $R_j^\top$ is the restriction operator $R_j : V_h' \to V_{h,0}(\Omega_j)'$ defined by $\langle R_j g, v\rangle = \langle g, R_j^\top v\rangle$, for $v \in V_{h,0}(\Omega_j)$, $g \in V_h'$. Let R_j be the matrix representation of R_j. This is a boolean matrix. Then the one level Additive Schwarz preconditioner is defined simply based on these interpolation operators as $M_{AS,1}^{-1} := \sum_{j=1}^N R_j^\top A_j^{-1} R_j$ where $A_j := R_j A R_j^\top$ are the local problem matrices.

In other words, the one level Schwarz preconditioner approximates the inverse of the global matrix A^{-1} by a sum of local inverses A_j^{-1}. The method is known to converge [11] as long as the subdomains and finite element spaces are chosen so that $V_h = \sum_{j=1}^N \left[R_j^\top V_{h,0}(\Omega_j)\right]$. In some sense this ensures that the local subdomains are *overlapping enough*. The drawback of the one level Schwarz method is that its convergence rate depends on the number of subdomains and thus scales poorly for large problems. The introduction of a coarse space is a, by now classical, way of weakening this dependence. Having chosen the coarse space V_H and an interpolation operator $R_H^\top : V_H \to V_h$, the two-level Additive Schwarz preconditioner is the most simple two level method: it reads

$$M_{AS,2}^{-1} := R_H^\top A_H^{-1} R_H + \sum_{j=1}^N R_j^\top A_j^{-1} R_j, \quad A_H := R_H A R_H^\top \text{ (Coarse problem matrix)}, \tag{2}$$

where R_H is the matrix representations of R_H.

The following theorem is simply a reformulation of the results in Chapter 2 of the book by Toselli and Widlund [11] where the abstract Schwarz theory is presented. We refer to there for the proof.

Theorem 1 (Condition number in the abstract Schwarz theory). *Let k_0 be the maximal degree of multiplicity of a point in Ω with respect to the partition into subdomains: $k_0 = \max_{x\in\Omega}\left(\#\{\Omega_j : 1 \le j \le N, x \in \overline{\Omega}_j\}\right)$.*

Assume that for a fixed constant C_0 there exists a stable splitting $(z_H, z_1, \ldots, z_N) \in V_H \times V_{h,0}(\Omega_1) \times \cdots \times V_{h,0}(\Omega_N)$ of any $v \in V_h$:

$$v = R_H^\top z_H + \sum_{j=1}^{N} R_j^\top z_j; \quad a(R_H^\top z_H, R_H^\top z_H) + \sum_{j=1}^{N} a(R_j^\top z_j, R_j^\top z_j) \le C_0^2 a(v,v). \tag{3}$$

Then the condition number of A preconditioned by the two level Additive Schwarz operator satisfies $\kappa\left(M_{AS,2}^{-1}A\right) \le (k_0+1)C_0^2$.

This theorem is the cornerstone of our method and we make our objective more precise thanks to these two remarks:

- The constant k_0 in the inequality does not depend on the number of subdomains but only on the geometry of the partition. For instance in two dimensions if a regular partition into rectangular subdomains is used then $k_0 = 4$ no matter what the total number of subdomains is. This means that the presence of k_0 in the estimate does not violate scalability.
- To make the theorem more precise, C_0^{-2} is a lower bound for the eigenvalues of the preconditioned operator and k_0+1 is an upper bound. The upper bound holds and is sharp regardless of the choice of the (non empty) coarse space. For this reason we do not work to improve the upper bound and instead we will work only on the lower bound through the stable splitting assumption.

Now the question of making the method robust with respect to the number of subdomains and the coefficients in the PDEs reduces to the following problem:

> Find a coarse space V_H for which there exists a constant C_0 independent of the number of subdomains and the coefficients in the underlying set of PDEs such that any $v \in V_h$ admits a stable splitting (3) onto this coarse space and the local subspaces.

2 From the abstract Schwarz theory to the GenEO coarse space

The practical inconvenience of the stable splitting property is that it is not local. Reducing it to N local problems relies on the following observation: there are two simple ways to get a local version of v, either with the restriction operator $R_j v$ which returns a function in $V_{h,0}(\Omega_j)$ that is supported in $\overline{\Omega_j}$ or by restricting the domain of v to Ω_j which we denote $v_{|\Omega_j}$. There is no immediate inequality between the global term $a(v,v)$ and any of the local terms $a_{\Omega_j}(R_j v, R_j v)$. However the alternative

inequality $a(v,v) \geq a_{\Omega_j}(v_{|\Omega_j}, v_{|\Omega_j})$ holds (and motivates the following lemma), since according to (1),

$$a(v,v) = a_\Omega(v,v) = a_{\Omega_j}(v_{|\Omega_j}, v_{|\Omega_j}) + \underbrace{a_{\Omega\setminus\Omega_j}(v_{|\Omega\setminus\Omega_j}, v_{|\Omega\setminus\Omega_j})}_{\geq 0}.$$

Lemma 1. *Given $v \in V_h$, if there exists a splitting $v = z_H + z_1 + \cdots + z_N$ such that each local component ($j = 1,\ldots,N$) satisfies $a(R_j^\top z_j, R_j^\top z_j) \leq C_1 a_{\Omega_j}(v_{|\Omega_j}, v_{|\Omega_j})$, then the splitting is stable in the sense of (3) for $C_0^2 = 2 + C_1 k_0(2k_0+1)$.*

Proof. Using the definition of k_0 we can bound the sum of the local contributions:

$$\sum_{j=1}^N a(R_j^\top z_j, R_j^\top z_j) \leq C_1 \sum_{j=1}^N a_{\Omega_j}(v_{|\Omega_j}, v_{|\Omega_j}) \leq C_1 k_0 a(v,v).$$

The bound for the energy of the coarse contribution follows from $R_H^\top z_H = v - \sum_{j=1}^N R_j^\top z_j$ which implies $a(R_H^\top z_H, R_H^\top z_H) \leq 2a(v,v) + 2a\left(\sum_{j=1}^N R_j^\top z_j, \sum_{j=1}^N R_j^\top z_j\right)$ and, by the definition of k_0 and the previous inequality,

$$a\left(\sum_{j=1}^N R_j^\top z_j, \sum_{j=1}^N R_j^\top z_j\right) \leq k_0 \sum_{j=1}^N a(R_j^\top z_j, R_j^\top z_j) \leq C_1 k_0^2 a(v,v). \tag{4}$$

Putting all of these estimates together ends the proof of the lemma. □

Lemma 1 also explains why we think of the coarse space as the space of *bad* components. Indeed, it states that it is enough to check that an estimate holds on each of the local components z_j of the splitting. Then this implies an estimate for the coarse component z_H and in turn the stable splitting assumption is satisfied.

An important tool in building the GenEO coarse space is a family of partition of unity operators. The particularity of these partition of unity operators is that they are defined at the degree of freedom level. The main consequence is that when the partition of unity is applied to a function we do not need to reinterpolate into the finite element space as is classically the case in partition of unity spaces where an application of the partition of unity is a multiplication by a continuous function.

Definition 1 (Partition of unity). *For each subdomain let $\mathrm{dof}(\Omega_j)$ be the set of degrees of freedom for which the associated basis function ϕ_k is supported in $\overline{\Omega_j}$: $\mathrm{dof}(\Omega_j) = \{k; supp(\phi_k) \subset \overline{\Omega_j}\}$. Then for each degree of freedom $k = 1,\ldots,n$ let $\{\mu_{j,k}\}_{\{j:k\in\mathrm{dof}(\Omega_j)\}}$ be a family of weights $\left(\mu_{j,k} \geq 1 \quad and \quad \sum_{\{j:k\in\mathrm{dof}(\Omega_j)\}} \frac{1}{\mu_{j,k}} = 1\right)$. Finally the local partition of unity operator for $v \in V_h$ written as $v = \sum_{k=1}^n v_k \phi_k$ is defined by*

$$\Xi_j(v_{|\Omega_j}) := \sum_{k\in\mathrm{dof}(\Omega_j)} \frac{1}{\mu_{j,k}} v_k \, \phi_{k|\Omega_j}.$$

This definition gives rise to a few remarks:

- A possible choice for the weights in the definition of the partition of unity is to use the multiplicity of each degree of freedom (this is what we use in the numerical section): for any degree of freedom k, $1 \leq k \leq n$, let μ_k denote the number of subdomains for which k is an internal degree of freedom, i.e.

$$\mu_k := \#\{j : 1 \leq j \leq N \text{ and } k \in \text{dof}(\Omega_j)\}.$$

  Then let $\mu_{j,k} = \mu_k$ for every subdomain j for which $k \in \text{dof}(\Omega_j)$.
- Other more coefficient adapted choices similar to those in [3] could be made.
- The family of operators $\{\Xi_j\}_{j=1,\ldots,N}$ indeed forms a partition of unity since $\sum_{j=1}^{N} R_j^\top \Xi_j(v_{|\Omega_j}) = v$ for any $v \in V_h$. This provides an obvious splitting of v onto the local subspaces.
- The partition of unity operator Ξ_j takes the restriction of a function to subdomain Ω_j and returns a function in $V_{h,0}(\Omega_j)$ (which is supported in $\overline{\Omega}_j$).
- If a degree of freedom k belongs to only one subdomain j then $\mu_{j,k} = 1$ and $\left(\Xi_j(v_{|\Omega_j})\right)_k = \left(v_{|\Omega_j}\right)_k$. This is the reason why the overlap plays a special role in the generalized eigenvalue problem which separates *good* and *bad* components. More detail is given in the proof of the final theorem.

Next we introduce the GenEO coarse space.

Definition 2 (GenEO coarse space).

(i) For each subdomain Ω_j ($1 \leq j \leq N$), let the overlap be given by

$$\Omega_j^\circ = \bigcup \{\tau \subset \overline{\Omega_j} : \exists\, j' \neq j \quad \textit{such that} \quad \tau \subset \overline{\Omega_{j'}}\}.$$

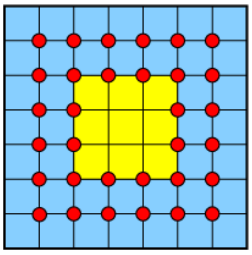
(ii) For each $j = 1,\ldots,N$, solve the following generalized eigenvalue problem: find the eigenpairs $(p_j^k, \lambda_j^k) \in \{v_{|\Omega_j}; v \in V_h\} \times \mathbb{R}^+$ of

$$a_{\Omega_j}(p_j^k, v_{|\Omega_j}) = \lambda_j^k\, a_{\Omega_j^\circ}(\Xi_j(p_j^k), \Xi_j(v_{|\Omega_j})) \quad \textit{for all} \ \ v \in V_h. \tag{5}$$

(iii) Given a threshold $\mathscr{K}_j$ for each $j = 1,\ldots,N$, let the GenEO coarse space be defined as

$$V_H := \text{span}\{R_j^\top \Xi_j(p_k^j) : \lambda_j^k \leq \mathscr{K}_j;\ j = 1,\ldots,N\}.$$

Assumption: An additional technical assumption is needed for the proof of Theorem 2. In [7] this is given rigorously in the abstract framework but here since we do not go into the details of the proof we will relie on the figure on the right. We assume that given data for the degrees of freedom in the overlap that do not lie on the boundary (i.e. the dots) we can build a discrete harmonic w.r.t. $a_{\Omega_j}(\cdot,\cdot)$ extension to the whole of Ω_j.



In the next theorem we give our main result which is an estimate for the condition number. It relies solely on the stable splitting property. We provide a suitable

decomposition that allows to complete the proof along with the main steps of the proof.

Theorem 2 (Stable Splitting and Final Estimate). *For any $j = 1, \dots, N$, suppose that the $p_j^k \in V_H$ have been normalized w.r.t. $a_{\Omega_j^\circ}(\Xi_j(\cdot), \Xi_j(\cdot))$ and let Π_j be the projection operator: $\Pi_j(v_{|\Omega_j}) = \sum_{\{k:\lambda_j^k \leq \mathscr{K}_j\}} a_{\Omega_j^\circ}(\Xi_j(p_j^k), \Xi_j(v_{|\Omega_j}))p_j^k$. Then, for any $v \in V_h$, the splitting $z_H := \sum_{j=1}^N \Xi_j\left(\Pi^j(v_{|\Omega_j})\right)$ and $z_j := \Xi_j\left(v_{|\Omega_j} - \Pi^j(v_{|\Omega_j})\right)$ satisfies Lemma 1 for $C_1 = \max_{1\leq j\leq N}\left(1 + \frac{1}{\mathscr{K}_j}\right)$ so, by Theorem 1, the condition number of the preconditioned operator is bounded by*

$$\kappa(M_{AS,2}^{-1}A) \;\leq\; (1+k_0)\left[2 + k_0(2k_0+1)\max_{1\leq j\leq N}\left(1+\frac{1}{\mathscr{K}_j}\right)\right],$$

Proof. The only thing that we need to check is $a(R_j^\top z_j, R_j^\top z_j) \leq \left(1 + \frac{1}{\mathscr{K}_j}\right)a(v,v)$. Here we only give the key ideas of the proof, the whole proof in a more general setting can be found in [7]. The most important ingredient in the proof is that, because they were obtained through a generalized eigenvalue problem, the p_j^k form a basis of $\{v_{|\Omega_j}; v \in V_h\}$ with the additional orthogonality type properties:

$$a_{\Omega_j^\circ}(\Xi_j(p_j^k), \Xi_j(p_j^l)) = 0 \quad \text{and} \quad a_{\Omega_j}(p_j^k, p_j^l) = 0 \quad \text{for all } k \neq l. \tag{6}$$

Using these properties we obtain

$$v_{|\Omega_j} - \Pi^j(v_{|\Omega_j}) = \sum_{\{k:\lambda_j^k > \mathscr{K}_j\}} \alpha_j^k p_j^k, \text{ for any } v_{|\Omega_j} \text{ written as } v_{|\Omega_j} = \sum_k \alpha_j^k p_j^k,$$

where the coefficients $\alpha_j^k \in \mathbb{R}$. Then we make appear the overlap term:

$$a(R_j^\top z_j, R_j^\top z_j) = a_{\Omega_j}(z_j, z_j) = a_{\Omega_j^\circ}(z_j, z_j) + a_{\Omega_j \setminus \Omega_j^\circ}(z_j, z_j).$$

In the interior $\Omega_j \setminus \Omega_j^\circ$ we have that Ξ_j is identity so $z_j = v_{|\Omega_j} - \Pi^j(v_{|\Omega_j})$ and because $a_{\Omega_j\setminus\Omega_j^\circ}(\cdot,\cdot) \leq a_{\Omega_j}(\cdot,\cdot)$: $a_{\Omega_j\setminus\Omega_j^\circ}(z_j,z_j) \leq a_{\Omega_j}(v_{|\Omega_j} - \Pi^j(v_{|\Omega_j}), v_{|\Omega_j} - \Pi^j(v_{|\Omega_j}))$. Then by an orthogonality argument $a_{\Omega_j\setminus\Omega_j^\circ}(z_j,z_j) \leq a_{\Omega_j}(v_{|\Omega_j}, v_{|\Omega_j})$.

For the other term, we write

$$\begin{aligned}
a_{\Omega_j^\circ}(z_j,z_j) &= a_{\Omega_j^\circ}\left(\sum_{\{k:\lambda_j^k>\mathscr{K}_j\}} \alpha_j^k \Xi_j(p_j^k), \sum_{\{k:\lambda_j^k>\mathscr{K}_j\}} \alpha_j^k \Xi_j(p_j^k)\right) \\
&= \sum_{\{k:\lambda_j^k>\mathscr{K}_j\}} {\alpha_j^k}^2 a_{\Omega_j^\circ}(\Xi_j(p_j^k), \Xi_j(p_j^k)) \quad \text{(Orthogonality (6))} \\
&\leq \frac{1}{\mathscr{K}_j} \sum_{\{k:\lambda_j^k>\mathscr{K}_j\}} {\alpha_j^k}^2 a_{\Omega_j}(p_j^k, p_j^k) \quad \text{(Definition of eigenproblem (5))}
\end{aligned}$$

$$\leq \frac{1}{\mathscr{K}_j} \sum_{\{\text{all } k\}} \alpha_j^{k^2} a_{\Omega_j}(p_j^k, p_j^k) = \frac{1}{\mathscr{K}_j} a_{\Omega_j}(v_{|\Omega_j}, v_{|\Omega_j}).$$

□

3 Numerical results

We run a simulation for the Darcy equation $-\nabla \cdot (\alpha \nabla v) = 1$ in $\Omega = [0,1]^2$ with homogeneous Dirichlet boundary conditions on the whole of $\partial\Omega$. The mesh is 200×200 square elements further subdivided into triangles and the finite element discretization uses standard $\mathbb{P}_1$ basis functions. All the finite element data is generated using Freefem++ [5]. The coefficient distribution is rather random since it is given by a QR code. This is shown on the left hand side of Figure 1 where in the yellow (or light) parts $\alpha = 1$ and in the pink (or dark) parts $\alpha = 1000$. The decomposition into subdomains is the 100 subdomain partition obtained *via* Metis [6] where we add one layer of overlap to each subdomains. This is plotted in the middle of Figure 1. The results are shown on the right hand side of Figure 1 where we have plotted the condition number versus the coarse space size for different values of the threshold K_j which is used to select modes for the coarse space. We observe that the coarse space grows roughly linearly with the threshold but the condition number stabilizes quickly. What this illustrates is that there is a good compromise to be found between the size of the coarse space and the efficiency of the method. An automatic optimal choice for $\mathscr{K}_j$ is a subject for future research. More thorough numerical experiments can be found in [7, 8] including three dimensional examples and results for elasticity.

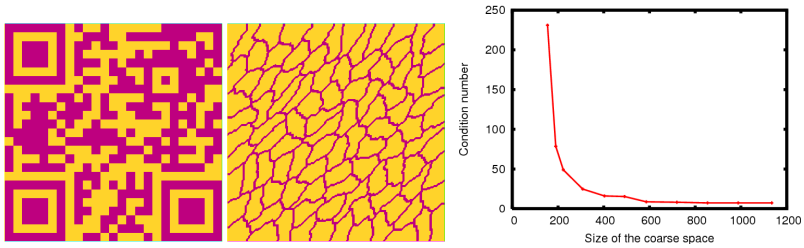


Fig. 1 Left: coefficient distribution (pink or dark is high conductivity) – Middle: Metis partition of the 200×200 mesh into 100 subdomains – Right: We plot the condition number with respect to the coarse space size when the threshold successively takes the values $\tau \in [0.01; 0.05; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9]$. As a matter of comparison: without any coarse space the condition number is 9661. With just the weighted constant $\Xi_j(1_{|\Omega_j})$ per floating subdomain the condition number is 7324: this 62 dimensional coarse space is what we get for GenEO with a barely positive threshold $\tau = 0^+$ (not shown on the graph simply because of scaling issues). We observe that the most troublesome eigenmodes are identified for quite a small value of the threshold and a reasonable size of the coarse space, then the condition number stagnates.

Conclusion

We have introduced the GenEO coarse space which is a way to automatically make the two level Schwarz method robust. The construction of this coarse space is based on solving generalized eigenvalue problems which isolate *good* and *bad* modes in each subdomain. We have presented the steps which lead to the choice of this generalized eigenvalue problem starting with the abstract Schwarz theory and the key ideas of the proof for the condition number estimate. The whole proof and a more general setting can be found in [7]. Although the eigenvalue problems are local, can be solved in parallel and only the smallest eigenvalues are needed, this setup phase could be costly and the study of the overall cost of the algorithm is still work in progress. The related methods in [2, 4] have been extended to a multilevel setting by [3, 12]. Moreover, this strategy was further applied by some of the authors in the BDD and FETI frameworks [9, 10].

References

1. Brezina, M., Heberton, C., Mandel, J., Vaněk, P.: An iterative method with convergence rate chosen a priori. Technical report University of Colorado Denver (1999). Earlier version presented at 1998 Copper Mountain Conference on Iterative Methods, April 1998.
2. Efendiev, Y., Galvis, J., Lazarov, R., Willems, J.: Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms. ESAIM: Mathematical Modelling and Numerical Analysis **46**(05), 1175–1199 (2012)
3. Efendiev, Y., Galvis, J., Vassilevski, P.: Multiscale spectral AMGe solvers for high-contrast flow problems. ISC-Preprint, Texas A&M University (2012). Submitted.
4. Galvis, J., Efendiev, Y.: Domain decomposition preconditioners for multiscale flows in high-contrast media. Multiscale Model. Simul. **8**(4), 1461–1483 (2010)
5. Hecht, F.: FreeFem++, 3rd edn. Numerical Mathematics and Scientific Computation. Laboratoire J.L. Lions, Université Pierre et Marie Curie, http://www.freefem.org/ff++/ (2012)
6. Karypis, G., Kumar, V.: METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. Department of Computer Science, University of Minnesota, http://glaros.dtc.umn.edu/gkhome/views/metis (1998)
7. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R.: Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. Technical report NuMa, Linz (2011). Submitted.
8. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R.: A robust two level domain decomposition preconditioner for systems of PDEs. Comptes Rendus Mathématique **349**(23-24), 1255–1259 (2011)
9. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Rixen, D.: Solving generalized eigenvalue problems on the interfaces to build a robust two level FETI method. HAL-00756840 (2012). Submitted.
10. Spillane, N., Rixen, D.: Automatic spectral coarse spaces for robust FETI and BDD algorithms. HAL-00756994 (2012). Submitted.
11. Toselli, A., Widlund, O.B.: Domain decomposition methods—algorithms and theory. Springer-Verlag, Berlin (2005)
12. Willems, J.: Robust multilevel methods for general symmetric positive definite operators. Technical report RICAM, Linz (2012). Submitted.

Optimized Schwarz algorithms in the framework of DDFV schemes

Martin J. Gander[1], Florence Hubert[2], and Stella Krell[3]

1 Introduction

We are interested in this paper in anisotropic diffusion problems of the form

$$-\mathrm{div}(A\nabla u) = f \text{ on } \Omega; \quad u = 0 \text{ on } \partial\Omega. \tag{1}$$

A discretization of the Schwarz algorithm using Discrete Duality Finite Volume methods (DDFV for short) for such problems was developed in [3]. The DDFV method needs a dual set of unknowns located on both vertices and "centers" of the primal control volumes, which leads to two meshes, the primal and the dual one, and permits the reconstruction of two-dimensional discrete gradients located on a third partition of Ω, called the diamond mesh, and also a discrete divergence operator defined by duality. The DDFV method is particularly accurate in terms of gradient approximation, see the benchmark [11] for problem (1) and an extensive bibliography. DDFV methods are also very robust, see [6, 2] for theoretical justifications, and [5] for applications. It is therefore of great interest to develop parallel solvers for such discretizations.

A non-overlapping Schwarz method using Robin transmission conditions was first proposed at the continuous level by Lions in [12]. For the model problem (1), the algorithm with two non-overlapping subdomains, $\Omega = \Omega_1 \cup \Omega_2$, and interface $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$, computes for iteration index $l \in \mathbb{N}^*$ the subdomain solutions

$$\begin{aligned} -\mathrm{div}(A\nabla u_j^l) &= f \quad \text{on } \Omega_j, \quad u_j^l = 0 \qquad \text{on } \partial\Omega_j \cap \partial\Omega, \\ A\nabla u_j^l \cdot \mathbf{n}_{ji} + p u_j^l &= -A\nabla u_i^{l-1}\mathbf{n}_{ij} + p u_i^{l-1} \quad \text{on } \Gamma,\ j \neq i, \end{aligned} \tag{2}$$

where $\mathbf{n}_{ji}$ is the unit normal from Ω_j to Ω_i, and p is a parameter that one can choose to accelerate convergence. Choosing p such that the algorithm converges as fast as possible leads to a so called optimized Schwarz method [8].

The non-overlapping algorithm ((2)) at the discrete level is interesting for coupling non-matching grids, see for example [1], [4] and [9] for isotropic diffusion problems or [10], [7] for general diffusion. It has also been analyzed in [3] in the case of highly anisotropic operators, and on a wide range of meshes. Numerical experiments in [3] showed however that the DDFV discretization chosen at the interfaces leads to a convergence factor of $1 - \mathcal{O}(h)$ of the algorithm (h denotes the

[1] University of Geneva, 2-4 rue du Lièvre CP 64 1211 Genève Switzerland e-mail: martin.gander@unige.ch ·[2] Aix-Marseille Université, LATP, 39 rue F. Joliot Curie 13 453 Marseille cedex 13, FRANCE e-mail: florence.hubert@univ-amu.fr ·[3] Université de Nice, Parc Valrose 28 avenue Valrose 06108 Nice Cedex 2 FRANCE e-mail: krell@unice.fr

mesh size), when the parameter p was chosen numerically such that convergence was fastest. This contraction factor is much worse than the optimal contraction factor $1-\mathcal{O}(\sqrt{h})$ of ((2)) for other discretizations, see [8]. The purpose of this short paper is to investigate why the classical DDFV discretization leads to such a slow convergence of the optimized Schwarz method, and to develop a new discretization of the transmission conditions in order to restore the optimal convergence rate. We show our results for the Poisson equation, $A = \mathrm{Id}$, but the extension to anisotropic tensors A can be obtained similarly. In Section (2), we show for the case of the Poisson equation and square meshes on half spaces that the traditional DDFV discretization leads to a mass matrix in the term with the Robin parameter. This mass matrix couples the primal and dual grids, and destroys the good convergence behavior of the optimized Schwarz method. In Section (3), we then show how to discretize the transmission conditions differently in the context of DDFV in order to recover the optimal convergence factor $1-\mathcal{O}(\sqrt{h})$. We then extend the algorithm to general meshes and prove convergence. Finally, in Section (4), we present numerical experiments which illustrate our analysis.

2 DDFV discretization of the optimized Schwarz algorithm

We decompose $\Omega := \mathbb{R}^2$ into two non-overlapping half planes $\Omega_1 := (-\infty, 0) \times \mathbb{R}$ and $\Omega_2 := (0, \infty) \times \mathbb{R}$, with the interface $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$. We use a regular grid of squares, so that the DDFV discretization away from the interface Γ leads to two interlaced five point finite difference schemes. The mesh size is denoted by h. We use for the scheme aligned with the interface star indices, and for the other one indices without stars, see Figure (1). The DDFV Schwarz algorithm proposed in [3] solves at each iteration $l \in \mathbb{N}^*$, on each domain j on interior primal cells

$$u^{j,l}_{m+1,n} - 2u^{j,l}_{m,n} + u^{j,l}_{m-1,n} + u^{j,l}_{m,n+1} - 2u^{j,l}_{m,n} + u^{j,l}_{m,n-1} = 0, m > 0. \tag{3}$$

In order to obtain (3) for $m = 1$, we introduce $u^{j,l}_{0,n}$ which is linked with the interface primal unknowns $u^{j,l}_{\frac{1}{2},n}$ by

$$u^{j,l}_{\frac{1}{2},n} = \frac{1}{2}(u^{j,l}_{1,n} + u^{j,l}_{0,n}). \tag{4}$$

On interior dual cells, the algorithm solves

$$u^{j,l}_{m^*+1,n^*} - 2u^{j,l}_{m^*,n^*} + u^{j,l}_{m^*-1,n^*} + u^{j,l}_{m^*,n^*+1} - 2u^{j,l}_{m^*,n^*} + u^{j,l}_{m^*,n^*-1} = 0, m^* > 0, \tag{5}$$

whereas on boundary dual cells, the additional fluxes $\varphi^{j,l}_{n,n^*}$ are used,

$$u^{j,l}_{1^*,n^*} - u^{j,l}_{0^*,n^*} + \frac{1}{2}(u^{j,l}_{0^*,n^*+1} - 2u^{j,l}_{0^*,n^*} + u^{j,l}_{0^*,n^*-1}) + \frac{h}{2}(\varphi^{j,l}_{n-1,n^*} + \varphi^{j,l}_{n,n^*}) = 0. \tag{6}$$

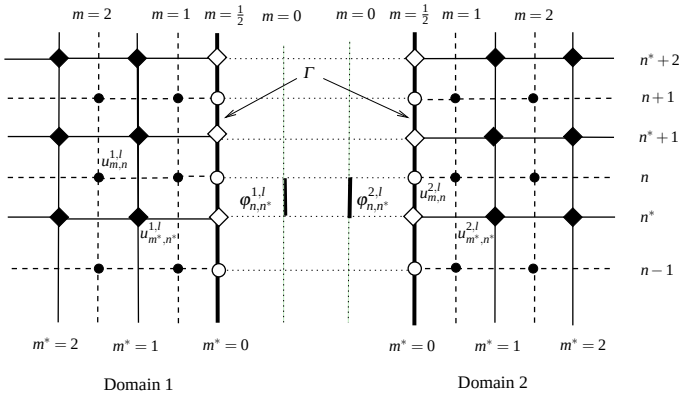


Fig. 1 The unknowns $u^{j,l}_{m,n}$ are associated with the primal cells, whose centers are bullets $\bullet$; the unknowns $u^{j,l}_{m^*,n^*}$ are associated with the dual cells shown in dashed, whose centers are diamonds $\blacklozenge$, or $\lozenge$ for boundary cells. The centers of the dual cells $\blacklozenge$ are the vertices of the primal cells, and similarly the centers of the primal cells $\bullet$ are the vertices of the dual cells. Additional primal unknowns $u^{j,l}_{\frac{1}{2},n}$, located at $\circ$, and also additional flux unknowns $\varphi^{j,l}_{n,n^*}$ are needed on the interface Γ. The indices j and l stand for the domain and the iteration.

The Robin transmission condition on Γ can now be expressed using the fluxes $\varphi^{j,l}_{n,n^*}$,

$$\varphi^{j,l}_{n,n^*} + \frac{p}{2}(u^{j,l}_{0^*,n^*} + u^{j,l}_{\frac{1}{2},n}) = -\varphi^{i,l-1}_{n,n^*} + \frac{p}{2}(u^{i,l-1}_{0^*,n^*} + u^{i,l-1}_{\frac{1}{2},n}). \tag{7}$$

Finally, a consistency condition is required for the fluxes, namely

$$\frac{1}{2}(\varphi^{j,l}_{n,n^*} + \varphi^{j,l}_{n,n^*+1}) = \frac{2}{h}(u^{j,l}_{\frac{1}{2},n} - u^{j,l}_{1,n}). \tag{8}$$

Equations (3)-(8) completely describe the original DDFV Schwarz algorithm from [3]. In order to analyze the DDFV discretization of the optimized Schwarz algorithm (3) and (5), we perform a discrete Fourier transform in the n index, which corresponds to the y variable, aligned with the interface. Setting $u^{j,l}_{m,n} = \hat{u}^{j,l}_{m,k} e^{iknh}$, $u^{j,l}_{m^*,n^*} = \hat{u}^{j,l}_{m^*,k} e^{ikn^*h}$, both $\hat{u}^{j,l}_{\cdot,k}$ and $\hat{u}^{j,l}_{(\cdot)^*,k}$ satisfy the recurrence relation

$$X_{m+1} - 2X_m + X_{m-1} + \alpha_k X_m = 0, \tag{9}$$

with $\alpha_k = 2\cos kh - 2$. The general solutions of (3) and (5) are bounded solutions of (9), which implies that

$$\hat{u}^{j,l}_{m,k} = C^{j,l}_k \lambda^m, \quad \hat{u}^{j,l}_{m^*,k} = C^{*,j,l}_k \lambda^{m^*}, \qquad \lambda := \frac{2-\alpha_k - \sqrt{(2-\alpha_k)^2-4}}{2}.$$

In order to determine the constants $C^{j,l}_k$ and $C^{*,j,l}_k$ from the transmission conditions (6) and (8), we eliminate the fluxes from the interface conditions using (7):

$$\frac{1}{h}(u^{j,l}_{0^*,n^*}-u^{j,l}_{1^*,n^*})-\frac{1}{2h}(u^{j,l}_{0^*,n^*+1}-2u^{j,l}_{0^*,n^*}+u^{j,l}_{0^*,n^*-1})+p\gamma_{n^*}(u^{j,l})$$
$$=-\frac{1}{h}(u^{i,l-1}_{0^*,n^*}-u^{i,l-1}_{1^*,n^*})+\frac{1}{2h}(u^{i,l-1}_{0^*,n^*+1}-2u^{i,l-1}_{0^*,n^*}+u^{i,l-1}_{0^*,n^*-1})+p\gamma_{n^*}(u^{i,l-1}),$$

and

$$\frac{2}{h}(u^{j,l}_{\frac{1}{2},n}-u^{j,l}_{1,n})+p\gamma_n(u^{j,l})=-\frac{2}{h}(u^{i,l-1}_{\frac{1}{2},n}-u^{i,l-1}_{1,n})+p\gamma_n(u^{i,l-1}),$$

with traces

$$\gamma_{n^*}(u^{j,l})=\frac{1}{4}(u^{j,l}_{\frac{1}{2},n}+2u^{j,l}_{0^*,n^*}+u^{j,l}_{\frac{1}{2},n-1}),\ \gamma_n(u^{j,l})=\frac{1}{4}(u^{j,l}_{0^*,n^*}+2u^{j,l}_{\frac{1}{2},n}+u^{j,l}_{0^*,n^*+1}). \quad (10)$$

We then obtain for the iteration of the constants using (4), $\begin{pmatrix} C_k^{j,l} \\ C_k^{*,j,l} \end{pmatrix} = B \begin{pmatrix} C_k^{i,l-1} \\ C_k^{*,i,l-1} \end{pmatrix}$ with the iteration matrix $B = M^{-1}N$, where

$$M=\begin{pmatrix} \frac{1}{h}(1-\lambda)+\frac{p}{4}(1+\lambda) & \frac{p}{4}(1+e^{ikh}) \\ \frac{p}{8}(1+\lambda)(1+e^{-ikh}) & \frac{1}{h}(1-\lambda)-\frac{\alpha_k}{2h}+\frac{p}{2} \end{pmatrix}$$
$$N=\begin{pmatrix} -\frac{1}{h}(1-\lambda)+\frac{p}{4}(1+\lambda) & \frac{p}{4}(1+e^{ikh}) \\ \frac{p}{8}(1+\lambda)(1+e^{-ikh}) & -\frac{1}{h}(1-\lambda)+\frac{\alpha_k}{2h}+\frac{p}{2} \end{pmatrix}.$$

Proposition 1. *The optimized parameter in the DDFV discretized Schwarz algorithm ((3)-(8)) satisfies* $p_{opt} = \mathrm{Argmin}_p \max_k(\rho(B)) = \frac{4}{h}$*, and the associated optimized contraction factor is* $1-\frac{1}{2}k_{\min}h+O(h^2)$.

Proof. The proof of this result is based on two observations: the minimum is obtained when both eigenvalues are the same, which is achieved with the given choice of p, and then the maximum is attained for the lowest mode $k = k_{\min}$. The computations are however too long and technical for this short paper.

3 A new DDFV Discretization of the Transmission Conditions

A careful comparison with the convergence results in [8] suggests that the mass matrices appearing in the traces $\gamma_n(u^{j,l})$ introduce an additional coupling, which prevents the optimized DDFV Schwarz algorithm from converging rapidly. Modifying the traces $\gamma_{n^*}(u^{j,l})$ in (10) to be lumped, i.e.

$$\gamma^{\mathrm{new}}_{n^*}(u^{j,l})=u^{j,l}_{0^*,n^*},\ \gamma^{\mathrm{new}}_n(u^{j,l})=u^{j,l}_{\frac{1}{2},n}, \quad (11)$$

the iteration matrix becomes diagonal: $B^{\mathrm{new}} = (M^{\mathrm{new}})^{-1}N^{\mathrm{new}}$, where

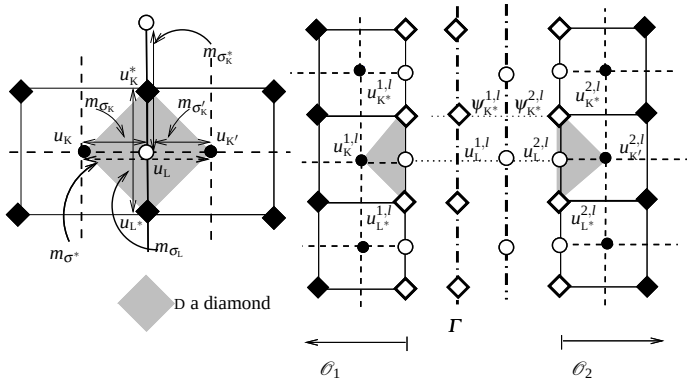


Fig. 2 Notation around a diamond. The new unknowns needed to describe the DDFV scheme on Ω as the limit of the Schwarz algorithm

$$M^{\text{new}} = \begin{pmatrix} \frac{1}{h}(1-\lambda)+\frac{p}{2}(1+\lambda) & 0 \\ 0 & \frac{1}{h}(1-\lambda)-\frac{\alpha_k}{2h}+p \end{pmatrix}$$
$$N^{\text{new}} = \begin{pmatrix} -\frac{1}{h}(1-\lambda)+\frac{p}{2}(1+\lambda) & 0 \\ 0 & -\frac{1}{h}(1-\lambda)+\frac{\alpha_k}{2h}+p \end{pmatrix},$$

and we obtain a much better convergence result.

Proposition 2. *The optimized parameter in the DDFV Schwarz algorithm ((3)-(8)) with modified traces* (11) *satisfies* $p_{opt} = \text{Argmin}_p \max_k(\rho(B^{new})) \sim \frac{2^{3/4}\sqrt{k_{\min}}}{\sqrt{h}}$, *and the associated optimized contraction factor is* $1 - 2^{1/4}\sqrt{k_{\min}}\sqrt{h} + O(h)$.

Proof. The proof of this result is based on equioscillation of the first eigenvalue of B^{new} at $k = k_{\min}$ and the second eigenvalue of B^{new} at $k = k_{\max} \approx \frac{\pi}{h}$, using asymptotic analysis. The details are however too long for this short paper.

We now describe the DDFV Schwarz algorithm for general subdomains and decompositions using the notation from [3]. DDFV schemes can be described by two operators: a discrete gradient $\nabla^{\mathfrak{D}}$ and a discrete divergence $(\text{div}_{\text{K}}, \text{div}_{\text{K}^*})$, which are dual to each other, see [2] or [3]. We refer to the primal unknowns by $u_{\text{K}}^{j,l}$ or $u_{\text{L}}^{j,l}$, to the dual unknowns by $u_{\text{K}^*}^{j,l}$ or $u_{\text{L}^*}^{j,l}$ and to the set of unknowns by $u^{j,l}$. The primal mesh on Ω_j is called $\mathfrak{M}_j$, the dual mesh on Ω_j is $\mathfrak{M}_j^*$ for the interior cells, $\partial\mathfrak{M}_{j,\Gamma}^*$ for the dual boundary cells related to Γ and the diamond mesh on Ω_j is called $\mathfrak{D}_j$. We further need additional unknowns $u_{\text{L}}^{j,l}$ on the edges of Γ denoted by $\partial\mathfrak{M}_{j,\Gamma}$, and additional fluxes $\psi_{\text{K}^*}^{j,l}$ for $\text{K}^* \in \partial\mathfrak{M}_{j,\Gamma}^*$ as shown in Figure (2). We denote by $\mathfrak{D}_{\text{K}^*}$ the set of diamonds such that $\text{D} \cap \text{K}^* \neq \emptyset$ for $\text{K}^* \in \partial\mathfrak{M}_{j,\Gamma}^*$. The DDFV Schwarz algorithm then computes for $l \in \mathbb{N}^*$, $j = 1,2$, $i = 2,1$

$$-\text{div}_{\text{K}}\left(\nabla^{\mathfrak{D}} u^{j,l}\right) = 0, \ \forall\, \text{K} \in \mathfrak{M}_j, \quad -\text{div}_{\text{K}^*}\left(\nabla^{\mathfrak{D}} u^{j,l}\right) = 0, \ \forall\, \text{K}^* \in \mathfrak{M}_j^*, \quad (12a)$$

$$-\sum_{\mathrm{D}\in\mathfrak{D}_{\mathrm{K}^*}} m_{\sigma^*}\left(\nabla^{\mathfrak{D}}u^{j,l},n_{\sigma_{\mathrm{K}^*}}\right)-m_{\sigma_{\mathrm{K}^*}}\psi^{j,l}_{\mathrm{K}^*}=0,\ \ \forall\,\mathrm{K}^*\in\partial\mathfrak{M}^*_{j,\Gamma}, \tag{12b}$$

$$\left(\nabla^{\mathfrak{D}}u^{j,l},n_{ji}\right)+pu^{j,l}_{\mathrm{L}}=-\left(\nabla^{\mathfrak{D}}u^{i,l-1},n_{ij}\right)+pu^{i,l-1}_{\mathrm{L}},\ \ \forall\,\mathrm{L}\in\partial\mathfrak{M}_{j,\Gamma}, \tag{12c}$$

$$\psi^{j,l}_{\mathrm{K}^*}+pu^{j,l}_{\mathrm{K}^*}=-\psi^{i,l-1}_{\mathrm{K}^*}+pu^{i,l-1}_{\mathrm{K}^*},\ \ \forall\,\mathrm{K}^*\in\partial\mathfrak{M}^*_{j,\Gamma}. \tag{12d}$$

Using the same discrete Fourier transform for (12) as in Section (2), we obtain B^{new}. Well-posedness of the algorithm can be proved using classical a priori estimates with the discrete duality property.

Theorem 1 (Convergence of the new Schwarz algorithm). *For all $p>0$, the solution of the new Schwarz algorithm* (12) *converges as l tends to infinity to the solution of the classical DDFV scheme for the Laplace equation on Ω.*

Proof. We first rewrite the classical DDFV scheme for the Laplace equation on Ω as the limit of the Schwarz algorithm. To this end, we introduce new unknowns near the boundary Γ, see Figure (2):

- for all $\mathrm{K}\in\mathfrak{M}_j$, we set $u^{j,\infty}_{\mathrm{K}}=u_{\mathrm{K}}$ and for all $\mathrm{K}^*\in\mathfrak{M}^*{}_j$, we set $u^{j,\infty}_{\mathrm{K}^*}=u_{\mathrm{K}^*}$,
- for all $\mathrm{L}\in\partial\mathfrak{M}_{j,\Gamma}$ choose $u^{j,\infty}_{\mathrm{L}}=u^{i,\infty}_{\mathrm{L}}=\frac{m_{\sigma_{\mathrm{K}'}}u_{\mathrm{K}}+m_{\sigma_{\mathrm{K}}}u_{\mathrm{K}'}}{m_{\sigma^*}}$,
- for all $\mathrm{K}^*\in\partial\mathfrak{M}^*_{j,\Gamma}$ choose $u^{j,\infty}_{\mathrm{K}^*}=u^{i,\infty}_{\mathrm{K}^*}=u_{\mathrm{K}^*}$ and

$$\psi^{j,\infty}_{\mathrm{K}^*}=-\psi^{i,\infty}_{\mathrm{K}^*}=-\frac{1}{m_{\sigma_{\mathrm{K}^*}}}\sum_{\mathrm{D}\in\mathfrak{D}_{\mathrm{K}^*}} m_{\sigma^*}\left(\nabla^{\mathfrak{D}}u^{j,\infty},n_{\sigma_{\mathrm{K}^*}}\right).$$

By linearity it suffices to prove the convergence of the new DDFV Schwarz algorithm (12) to zero. An a priori estimate using discrete duality leads to

$$2\sum_{\mathrm{D}\in\mathfrak{D}_j} m_{\mathrm{D}}\|\nabla^{\mathfrak{D}}u^{j,l+1}\|^2-\sum_{\mathrm{L}\in\partial\mathfrak{M}_{j,\Gamma}} m_{\sigma_{\mathrm{L}}}(\nabla^{\mathfrak{D}}u^{j,l+1},n_{\sigma_{\mathrm{L}}})u^{j,l+1}_{\mathrm{L}}-\sum_{\mathrm{K}^*\in\partial\mathfrak{M}^*_{j,\Gamma}} m_{\sigma_{\mathrm{K}^*}}\psi^{j,l+1}_{\mathrm{K}^*}u^{j,l+1}_{\mathrm{K}^*}=0.$$

We now rewrite the last two terms as

$$\begin{aligned}-\sum_{\mathrm{L}\in\partial\mathfrak{M}_{j,\Gamma}} m_{\sigma_{\mathrm{L}}}(\nabla^{\mathfrak{D}}u^{j,l+1},n_{\sigma_{\mathrm{L}}})u^{j,l+1}_{\mathrm{L}}&=\frac{1}{4p}\sum_{\mathrm{L}\in\partial\mathfrak{M}_{j,\Gamma}} m_{\sigma_{\mathrm{L}}}\left(-(\nabla^{\mathfrak{D}}u^{j,l+1},n_{\sigma_{\mathrm{L}}})+pu^{j,l+1}_{\mathrm{L}}\right)^2\\&\quad-\frac{1}{4p}\sum_{\mathrm{L}\in\partial\mathfrak{M}_{i,\Gamma}} m_{\sigma_{\mathrm{L}}}\left(-(\nabla^{\mathfrak{D}}u^{i,l},n_{\sigma_{\mathrm{L}}})+pu^{i,l}_{\mathrm{L}}\right)^2,\end{aligned}$$

and using (12b)

$$\begin{aligned}&-\sum_{\mathrm{K}^*\in\partial\mathfrak{M}^*_{j,\Gamma}} m_{\sigma_{\mathrm{K}^*}}\psi^{j,l+1}_{\mathrm{K}^*}u^{j,l+1}_{\mathrm{K}^*}\\&\quad=\frac{1}{4p}\sum_{\mathrm{K}^*\in\partial\mathfrak{M}^*_{j,\Gamma}} m_{\sigma_{\mathrm{K}^*}}\left(pu^{j,l+1}_{\mathrm{K}^*}-\psi^{j,l+1}_{\mathrm{K}^*}\right)^2-\frac{1}{4p}\sum_{\mathrm{K}^*\in\partial\mathfrak{M}^*_{i,\Gamma}} m_{\sigma_{\mathrm{K}^*}}\left(pu^{i,l}_{\mathrm{K}^*}-\psi^{i,l}_{\mathrm{K}^*}\right)^2.\end{aligned}$$

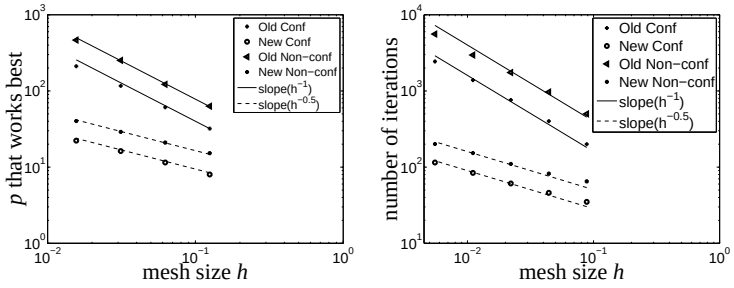


Fig. 3 Asymptotic behavior of the numerically optimized parameter p on the left, and number of iterations to reduce the error by a factor of 10^{-10} on the right

Summing over $l = 0, \cdots, l_{max}-1$ and $j = 1,2$, we get

$$
\begin{aligned}
2 &\sum_{l=0}^{l_{max}-1} \sum_{j=1,2} \sum_{\mathrm{D}\in\mathfrak{D}_j} m_{\mathrm{D}} \|\nabla^{\mathfrak{D}} u^{j,l+1}\|^2 + \frac{1}{4p} \sum_{j=1,2} \sum_{\mathrm{L}\in\partial\mathfrak{M}_{j,\Gamma}} m_{\sigma_{\mathrm{L}}} \Big(p u_{\mathrm{L}}^{j,l_{max}} - (\nabla^{\mathfrak{D}} u^{j,l_{max}}, n_{\sigma_{\mathrm{L}}}) \Big)^2 \\
&+ \frac{1}{4p} \sum_{j=1,2} \sum_{\mathrm{K}^*\in\partial\mathfrak{M}^*_{j,\Gamma}} m_{\sigma_{\mathrm{K}^*}} \left(-\psi_{\mathrm{K}^*}^{j,l_{max}} + p u_{\mathrm{K}^*}^{j,l_{max}} \right)^2 \\
&= \sum_{j=1,2} \frac{1}{4p} \left(\sum_{\mathrm{L}\in\partial\mathfrak{M}_{j,\Gamma}} m_{\sigma_{\mathrm{L}}} \Big(-(\nabla^{\mathfrak{D}} u^{j,0}, n_{\sigma_{\mathrm{L}}}) + p u_{\mathrm{L}}^{j,0} \Big)^2 + \sum_{\mathrm{K}^*\in\partial\mathfrak{M}^*_{j,\Gamma}} m_{\sigma_{\mathrm{K}^*}} \Big(-\psi_{\mathrm{K}^*}^{j,0} + p u_{\mathrm{K}^*}^{j,0} \Big)^2 \right).
\end{aligned}
$$

This shows that the total energy stays bounded as the iteration l goes to infinity, and hence the algorithm converges.

4 Numerical experiments

We show results for Laplace's equation on $\Omega = (-1,1)^2$ with two subdomains $x > 0$ and $x < 0$. We first simulate in Figure (3) the error equations, i.e. using homogeneous data, and starting with a random initial guess. On the left, we show the p that worked best as h is refined, both for a conforming square mesh ($2^i \times 2^i$ squares on Ω_j, $j = 1,2$), and for a non-conforming square mesh ($2^i \times 2^i$ squares on Ω_1 and $3^i \times 3^i$ squares on Ω_2). On the right, we show the number of iterations needed to get an error reduction of 10^{-10}. These experiments illustrate well our theoretical results.

We next show a case with exact solution $u(x,y) = \cos(2.5\pi x)\cos(2.5\pi y)$. Starting with a random initial guess, Figure (4) shows the convergence history of the algorithms for various parameters p on the left, and snapshots of the error at iteration 10 on the right. We clearly see that for p too small, high frequencies dominate the error, and for p large low frequencies. In the old algorithm, the theoretically optimized choice $p = 90.5$, and in the new algorithm the theoretically optimized choice $p = 14.18$ will work best in the long run. Finally, a priori knowledge of the

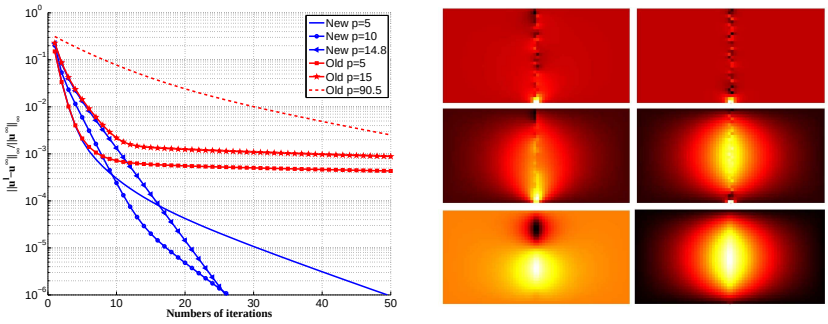


Fig. 4 Left: convergence history on a conforming 32×32 square mesh. Right: snapshots of the error at iteration 10, left column for the old version and $p = 5, 15, 90.5$, right column for the new version and $p = 5, 10, 14.18$

frequency content of the solution can be used to choose a p that gives very rapid convergence early on in the iteration (here $p = 5$, good for low frequencies). This choice becomes however very bad in the long run, once other error frequencies become important.

References

1. Achdou, Y., Japhet, C., Nataf, F., Maday, Y.: A new cement to glue non-conforming grids with Robin interface conditions: The finite volume case. Numer. Math. **92**(4), 593–620 (2002)
2. Andreianov, B., Boyer, F., Hubert, F.: Discrete duality finite volume schemes for Leray-Lions type elliptic problems on general 2D-meshes. Num. Meth. for PDEs **23**(1), 145–195 (2007)
3. Boyer, F., Hubert, F., Krell, S.: Non-overlapping Schwarz algorithm for solving 2d m-DDFV schemes. IMA Jour. Num. Anal. **30** (2009)
4. Cautrès, R., Herbin, R., Hubert, F.: The Lions domain decomposition algorithm on non-matching cell-centred finite volume meshes. IMA J. Numer. Anal. **24**(3), 465–490 (2004)
5. Coudière, Y., Pierre, C.: Stability and convergence of a finite volume method for two systems of reaction-diffusion equations in electro-cardiology. Nonlinear Anal. R. World Appl. **7**(4), 916–935 (2006)
6. Domelevo, K., Omnes, P.: A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids. M2AN Math. Model. Numer. Anal. **39**(6), 1203–1249 (2005)
7. Dubois, O.: Optimized Schwarz methods for the advection-diffusion equation and for problems with discontinuous coefficients. Ph.D. thesis, McGill University, Canada (June 2007)
8. Gander, M.J.: Optimized Schwarz method. SIAM J. on Numer. Anal. **44**(2), 699–731 (2006)
9. Gander, M.J., Japhet, C., Maday, Y., Nataf, F.: A new cement to glue nonconforming grids with Robin interface conditions: the finite element case. Lect. Notes Comput. Sci. Eng. **40**, 259–266 (2005)
10. Gerardo-Giorda, L., Nataf, F.: Optimized schwarz methods for unsymmetric layered problems with strongly discontinuous and anisotropic coefficients. J. Numer. Math. **13**, 265–294 (2005)
11. Herbin, R., Hubert, F.: Benchmark on discretization schemes for anisotropic diffsion problems on general grids. In: R. Eymard, J.M. Hérard (eds.) Proceedings of FVCA V. Hermès (2008)
12. Lions, P.L.: On the Schwarz alternating method. III. A variant for nonoverlapping subdomains. In: Third International Symposium on DDM (Houston, 1989), pp. 202–223. SIAM (1990)

A Time-Dependent Dirichlet-Neumann Method for the Heat Equation

Bankim C. Mandal[1]

1 Introduction

We introduce a new Waveform Relaxation (WR) method based on the Dirichlet-Neumann algorithm and present convergence results for it in one space dimension. To solve time-dependent problems in parallel, one can either discretize in time to obtain a sequence of steady problems to which the domain decomposition algorithms are applied, or apply WR to the large system of ordinary differential equations (ODEs) obtained from spatial discretization. The credit of WR method goes to Picard [14] and Lindelöf [9] for the solution of ODEs in the late 19th century. Lelarasmee, Ruehli and Sangiovanni-Vincentelli [8] were the first to introduce the WR as a parallel method for the solution of ODEs. The main advantage of the WR method is that one can use different time steps in different space-time subdomains. The authors of [6] and [7] then generalized WR methods for ODEs to solve time-dependent PDEs. Gander and Stuart [6] showed linear convergence of overlapping Schwarz WR iteration for the heat equation on unbounded time intervals with a rate depending on the size of the overlap; Giladi and Keller [7] proved superlinear convergence of the Schwarz WR method with overlap for the convection-diffusion equation on bounded time intervals.

The Dirichlet-Neumann method, which belongs to the class of substructuring methods, is based on a non-overlapping spatial domain decomposition. The iteration involves subdomain solves with Dirichlet boundary conditions, followed by subdomain solves with Neumann boundary conditions. The Dirichlet-Neumann algorithm was first considered for elliptic problems by P. E. Bjørstad & O. Widlund [1] and further discussed in [2], [11] and [12]. In this paper, we propose the Dirichlet-Neumann Waveform Relaxation (DNWR) method, a new Dirichlet-Neumann analogue of WR for the time-dependent problems. For presentation purposes, we derive our results for two subdomains in one spatial dimension. We discuss the method in the continuous setting to ensure the understanding of the asymptotic behavior of the method in the case of fine grids.

We consider the following initial boundary value problem (IBVP) for the heat equation as our guiding example on a bounded domain $\Omega \subset \mathbb{R}, 0 < t < T$,

$$\begin{aligned} \tfrac{\partial u}{\partial t} &= \Delta u + f(x,t), \; x \in \Omega, 0 < t < T, \\ u(x,0) &= u_0(x), \qquad x \in \Omega, \\ u(x,t) &= g(x,t), \qquad x \in \partial\Omega, 0 < t < T. \end{aligned} \tag{1}$$

[1] Department of Mathematics, University of Geneva, Geneva, Switzerland, e-mail: Bankim.Mandal@unige.ch

2 The Dirichlet-Neumann Waveform Relaxation algorithm

To define the Dirichlet-Neumann iterative method for the model problem (1) on the domain $(-b,a)\times(0,T)$, we split the spatial domain $\Omega=(-b,a)$ into two non-overlapping subdomains, the Dirichlet subdomain $\Omega_1=(-b,0)$ and the Neumann subdomain $\Omega_2=(0,a)$, for $0<a,b<\infty$. The Dirichlet-Neumann Waveform Relaxation algorithm consists of the following steps: given an initial guess $h^0(t), t\in(0,T)$ along the interface $\Gamma=\{x=0\}$ and for $k=0,1,2,\ldots$, do

$$\begin{cases} \partial_t u_1^{k+1}-\partial_{xx}u_1^{k+1}=f(x,t), & x\in\Omega_1,\\ u_1^{k+1}(x,0)=u_0(x), & x\in\Omega_1,\\ u_1^{k+1}(-b,t)=g(-b,t),\\ u_1^{k+1}(0,t)=h^k(t), \end{cases} \quad \begin{cases} \partial_t u_2^{k+1}-\partial_{xx}u_2^{k+1}=f(x,t), & x\in\Omega_2,\\ u_2^{k+1}(x,0)=u_0(x), & x\in\Omega_2,\\ \partial_x u_2^{k+1}(0,t)=\partial_x u_1^{k+1}(0,t),\\ u_2^{k+1}(a,t)=g(a,t), \end{cases} \tag{2}$$

with the updating condition

$$h^{k+1}(t)=\theta u_2^{k+1}(0,t)+(1-\theta)h^k(t), \tag{3}$$

θ being a positive relaxation parameter. The parameter θ is chosen in $(0,1]$ to accelerate convergence. As the main goal of the analysis is to study how the error $h^k(t)-u(0,t)$ converges to zero, by linearity it suffices to consider the homogeneous problem, $f(x,t)=0$, $g(x,t)=0$, $u_0(x)=0$ in (1), and examine how $h^k(t)$ goes to zero as $k\to\infty$.

3 Convergence analysis and main results

We analyze the DNWR algorithm using the Laplace transform method. The Laplace transform of a function $w(t)$, defined for all real numbers $t\in[0,\infty)$, is the function $\hat{w}(s)$, defined by

$$\hat{w}(s)=\mathscr{L}\{w(t)\}:=\int_0^\infty e^{-st}w(t)\,dt,$$

(if the integral exists) s being a complex variable. If $\mathscr{L}\{w(t)\}=\hat{w}(s)$, then the inverse Laplace transform of $\hat{w}(s)$ is denoted by

$$\mathscr{L}^{-1}\{\hat{w}(s)\}:=w(t),\qquad t\geq 0,$$

which maps the Laplace transform of a function back to the original function. For more information on Laplace transforms, see [3, 13]. We use hats to denote the Laplace transform of a function in time in the rest of the paper.

Analysis by Laplace transforms. Applying a Laplace transform in time to (2) and solving the resulting ODEs yields the solutions: $\hat{u}_1^{k+1}(x,s)=\frac{\hat{h}^k(s)}{\sinh(b\sqrt{s})}\sinh\{(x+b)\sqrt{s}\}$

and $\hat{u}_2^{k+1}(x,s) = \hat{h}^k(s)\frac{\coth(b\sqrt{s})}{\cosh(a\sqrt{s})}\sinh\{(x-a)\sqrt{s}\}$. Now, evaluating $\hat{u}_2^{k+1}(x,s)$ at $x=0$ and inserting it into the transformed updating condition (3), we get for $k=0,1,2,\ldots$ $\hat{h}^{k+1}(s) = \{1-\theta-\theta\tanh(a\sqrt{s})\coth(b\sqrt{s})\}\hat{h}^k(s)$. Therefore, by induction we get

$$\hat{h}^k(s) = \left\{1-\theta-\theta\tanh(a\sqrt{s})\coth(b\sqrt{s})\right\}^k \hat{h}^0(s), \quad k=1,2,3,... \tag{4}$$

Theorem 1. *For the symmetric case, $a=b$ in (2)-(3), the DNWR algorithm converges linearly for $0<\theta<1$. Moreover, for $\theta=0.5$, it converges to the exact solution in two iterations, independent of the size of the time window.*

Proof. For $a=b$, the equation (4) reduces to $\hat{h}^k(s) = (1-2\theta)^k\hat{h}^0(s)$, which upon back transforming gives $h^k(t) = (1-2\theta)^k h^0(t)$. Thus, the convergence is linear for $\theta \neq 0.5$. On the other hand, for $\theta = 0.5$, $h^1(t) = 0$. Therefore, one more iteration produces the desired solution on the whole domain. □

The main area of concern for the rest of the paper is the analysis of the DNWR algorithm for $a \neq b$. If we define

$$G(s) := \tanh(a\sqrt{s})\coth(b\sqrt{s}) - 1 = \frac{\sinh((a-b)\sqrt{s})}{\cosh(a\sqrt{s})\sinh(b\sqrt{s})},$$

then the recurrence relation (4) reduces to

$$\hat{h}^k(s) = \begin{cases} \{q(\theta)-\theta G(s)\}^k\hat{h}^0(s), & \theta \neq 1/2 \\ (-1)^k 2^{-k} G^k(s)\hat{h}^0(s), & \theta = 1/2, \end{cases} \tag{5}$$

where $q(\theta) = 1-2\theta$. Note that for $\mathrm{Re}(s)>0$, $G(s)$ is[1] $\mathcal{O}(s^{-p})$ for every positive p. Therefore, by [3, p. 178], $G(s)$ is the Laplace transform of an analytic function $F_1(t)$ (in fact this is the motivation in defining G). In general, define $F_k(t) := \mathscr{L}^{-1}\left\{G^k(s)\right\}$ for $k=1,2,3,\ldots$. For θ not equal to $1/2$, h^k cannot be expressed as a simple convolution of h^0 and an analytic function; thus, different techniques are required to analyze its behavior. This case will be treated in a future paper. For $\theta = 1/2$ and $t \in (0,T)$ we get from (5)

$$\left|h^k(t)\right| = \left|2^{-k}\int_0^t (-1)^k h^0(t-\tau)F_k(\tau)d\tau\right| \leq 2^{-k} \parallel h^0 \parallel_{L^\infty(0,T)} \int_0^T \left|F_k(\tau)\right| d\tau. \tag{6}$$

So, we need to bound $\int_0^T \left|F_k(\tau)\right| d\tau$ to get an L^∞ convergence estimate. We concentrate on showing that $F_1(t)$ does not change signs both for the case $b<a$, in which $F_1(t) \geq 0$, and for $b \geq a$, for which $F_1(t) \leq 0$. Before we proceed further with the proof we need the following lemmas.

Lemma 1. *Let, $w(t)$ be a continuous and L^1-integrable function on $(0,\infty)$ with $w(t) \geq 0$ for all $t \geq 0$. Assume $W(s) = \mathscr{L}\{w(t)\}$. Then, for $\tau > 0$,*

[1] Assuming $s = re^{i\vartheta}$, $z = \sqrt{s}$, we can write for $b \geq a$, $\left|s^p G(s)\right| \leq \left|\frac{s^p}{\cosh(az)}\right| \leq \frac{2r^p}{\left|e^{a\sqrt{r/2}} - e^{-a\sqrt{r/2}}\right|} \to 0$, as $r \to \infty$; and for $a > b$, $\left|s^p G(s)\right| \leq \left|\frac{s^p}{\sinh(bz)}\right| \leq \frac{2r^p}{\left|e^{b\sqrt{r/2}} - e^{-b\sqrt{r/2}}\right|} \to 0$, as $r \to \infty$.

$$\int_0^\tau |w(t)|dt \le \lim_{s\to 0+} W(s).$$

Proof. Using the definition of Laplace transform, we have

$$\begin{aligned}\int_0^\tau |w(t)|dt &= \int_0^\tau w(t)dt \le \int_0^\infty w(t)dt \\ &= \int_0^\infty \lim_{s\to 0+} e^{-st}w(t)dt = \lim_{s\to 0+}\int_0^\infty e^{-st}w(t)dt \text{ (by Dominated Conv. Theorem)} \\ &= \lim_{s\to 0+} W(s). \qquad \square\end{aligned}$$

Lemma 2. *Let $\beta > \alpha \ge 0$ and s be a complex variable. Then, for $t \in (0,\infty)$*

$$\varphi(t) := \mathscr{L}^{-1}\left\{\frac{\sinh(\alpha\sqrt{s})}{\sinh(\beta\sqrt{s})}\right\} \ge 0\,;\ \ \psi(t) := \mathscr{L}^{-1}\left\{\frac{\cosh(\alpha\sqrt{s})}{\cosh(\beta\sqrt{s})}\right\} \ge 0.$$

Proof. First, let us consider the following IBVP for the heat equation on $(0,\beta)$: $u_t - u_{xx} = 0$, $u(x,0) = 0$, $u(0,t) = 0$, $u(\beta,t) = g(t)$. Therefore, for g non-negative, $u(\alpha,t)$ is also non-negative for all $t > 0$, thanks to the maximum principle. Now using the Laplace transform method, we get the solution along $x = \alpha$ as

$$\hat{u}(\alpha,s) = \hat{g}(s)\frac{\sinh(\alpha\sqrt{s})}{\sinh(\beta\sqrt{s})} \quad \Longrightarrow \quad u(\alpha,t) = \int_0^t g(t-\tau)\varphi(\tau)d\tau.$$

We prove the result by contradiction: suppose $\varphi(t_0) < 0$ for some $t_0 > 0$. Then by continuity of φ, there exists $\delta > 0$ such that $\varphi(\tau) < 0$, for $\tau \in (t_0-\delta, t_0+\delta)$. Now for $t > t_0 + \delta$, we choose g as

$$g(\varsigma) = \begin{cases} 1, & \varsigma \in (t-t_0-\delta, t-t_0+\delta) \\ 0, & \text{else.}\end{cases}$$

Then $u(\alpha,t) = \int_{t_0-\delta}^{t_0+\delta} g(t-\tau)\varphi(\tau)d\tau = \int_{t_0-\delta}^{t_0+\delta} \varphi(\tau)d\tau < 0$, a contradiction. This proves φ to be non-negative. For ψ, applying the Laplace transform method to the IBVP for the heat equation $u_t - u_{xx} = 0$, $u(x,0) = 0$, $u(-\beta,t) = g(t)$, $u(\beta,t) = g(t)$ yields the solution along $x = \alpha$ as: $\hat{u}(\alpha,s) = \hat{g}(s)\frac{\cosh(\alpha\sqrt{s})}{\cosh(\beta\sqrt{s})}$. Thus, a similar argument as in the first case proves that ψ is also non-negative. $\square$

Theorem 2. ***(Linear convergence bound for the Heat equation)*** *Let $\theta = 1/2$. For $T > 0$, the error of the Dirichlet-Neumann Waveform Relaxation (DNWR) algorithm satisfies*

$$\| h^k \|_{L^\infty(0,T)} \le \left(\frac{|b-a|}{2b}\right)^k \| h^0 \|_{L^\infty(0,T)}\,.$$

We therefore have a contraction if $a < 3b$.

Proof. By virtue of (6), it is sufficient to bound $\int_0^T \left|F_k(\tau)\right| d\tau$ for both $b \ge a$ and $a > b$, where $F_k(t) = \mathscr{L}^{-1}\left\{G^k(s)\right\}$. Suppose $b \ge a > 0$. We have $\mathscr{L}\{-F_1(t)\} =$

$\frac{\sinh((b-a)\sqrt{s})}{\sinh(b\sqrt{s})} \cdot \frac{1}{\cosh(a\sqrt{s})}$. So by Lemma 2 and the fact that the convolution of two positive functions is positive, $-F_1(t)$ is positive. Thus, by induction and with the same arguments, $(-1)^k F_k(t) \geq 0$ for all t. Therefore by Lemma 1

$$\int_0^T \left|(-1)^k F_k(\tau)\right| d\tau \leq \lim_{s\to 0+} (-1)^k G^k(s) = \left(\frac{b-a}{b}\right)^k. \tag{7}$$

Now let $a > b > 0$. We claim that $F_1(t)$ is positive. If $a - b \leq b$, then we get the positivity by Lemma 2. If this is not the case, then take the integer $m = \lfloor a/b \rfloor$ so that $mb < a \leq (m+1)b$. Then, recursively applying the identity

$$\frac{\sinh((a-jb)\sqrt{s})}{\sinh(b\sqrt{s})} = \frac{\sinh((a-(j+1)b)\sqrt{s})}{\sinh(b\sqrt{s})}\cosh(b\sqrt{s}) + \cosh((a-(j+1)b)\sqrt{s})$$

for $j = 1, \ldots, m-1$, we obtain

$$\frac{\sinh((a-b)\sqrt{s})}{\cosh(a\sqrt{s})\sinh(b\sqrt{s})} = \frac{\sinh((a-mb)\sqrt{s})}{\sinh(b\sqrt{s})} \cdot \frac{\cosh^{m-1}(b\sqrt{s})}{\cosh(a\sqrt{s})} + \sum_{j=0}^{m-2} \frac{\cosh^j(b\sqrt{s})\cosh((a-(j+2)b)\sqrt{s})}{\cosh(a\sqrt{s})}.$$

Applying the binomial theorem to $\cosh\theta = \left(e^{\theta} + e^{-\theta}\right)/2$ we have the power-reduction formula

$$\cosh^n\theta = \begin{cases} \frac{2}{2^n} \sum_{l=0}^{\frac{n-1}{2}} \binom{n}{l} \cosh\left((n-2l)\theta\right), & n \text{ odd}, \\ \frac{1}{2^n}\binom{n}{n/2} + \frac{2}{2^n} \sum_{l=0}^{\frac{n}{2}-1} \binom{n}{l} \cosh\left((n-2l)\theta\right), & n \text{ even}, \end{cases}$$

so that we can write $\cosh^n\theta = \sum_{l=0}^{n} A_l^n \cosh(l\theta)$ with $\sum_{l=0}^{n} A_l^n = 1$ and $A_l^n \geq 0$. Therefore, we have

$$G(s) = \frac{\sinh((a-b)\sqrt{s})}{\cosh(a\sqrt{s})\sinh(b\sqrt{s})} = \frac{\sinh((a-mb)\sqrt{s})}{\sinh(b\sqrt{s})} \sum_{l=0}^{m-1} A_l^{m-1} \frac{\cosh(lb\sqrt{s})}{\cosh(a\sqrt{s})} + \sum_{j=0}^{m-2}\sum_{l=0}^{j} \frac{A_l^j}{2} \left\{ \frac{\cosh\left((a-(j+l+2)b)\sqrt{s}\right)}{\cosh(a\sqrt{s})} + \frac{\cosh\left((a-(j-l+2)b)\sqrt{s}\right)}{\cosh(a\sqrt{s})} \right\},$$

where $\cosh^j\theta = \sum_{l=0}^{j} A_l^j \cosh(l\theta)$. Note that $a - mb \leq b$, $(j-l+2)b \leq mb < a$ and $|a-(j+l+2)b| < a$ for $0 \leq j, l \leq m-2$ and cosh is an even function. Thus by Lemma 2, each term in the above expression is the Laplace transform of a posi-

tive function. Hence $F_1(t)$ is positive, and therefore the convolution of k F_1's (i.e. $F_k(t)$) is also positive. We have $\lim_{s\to 0+} G(s) = \lim_{s\to 0+} \frac{\sinh((a-b)\sqrt{s})}{\cosh(a\sqrt{s})\sinh(b\sqrt{s})} = \frac{a-b}{b}$, and so by Lemma 1

$$\int_0^T \left|F_k(\tau)\right| d\tau = \int_0^T F_k(\tau) d\tau \leq \lim_{s\to 0+} G^k(s) = \left(\lim_{s\to 0+} G(s)\right)^k = \left(\frac{a-b}{b}\right)^k. \quad (8)$$

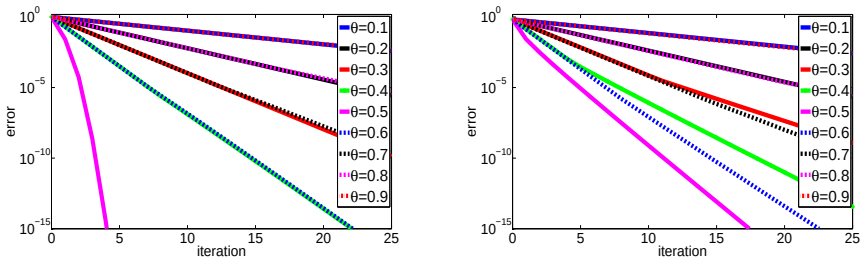
The result follows by inserting the estimates (7) and (8) into (6). □

4 Numerical Experiments

We perform experiments to measure the actual convergence rate of the DNWR algorithm for the problem

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = -e^{-t-x^2}, & x \in (-3,2), \\ u(x,0) = e^{-2x}, & x \in (-3,2), \\ u(-3,t) = e^{-2t} = u(2,t), & t > 0. \end{cases}$$

To solve the equation using the Dirichlet-Neumann algorithm, we discretize the Laplacian using centered finite differences in space and backward Euler in time on a grid with $\Delta x = 2 \times 10^{-2}$ and $\Delta t = 4 \times 10^{-4}$. For the numerical experiments we split the spatial domain into two non-overlapping subdomains $[-3,0]$ and $[0,2]$, so that $b = 3$ and $a = 2$ in (2)-(3). Thus this is the case when the Dirichlet subdomain is bigger than the Neumann subdomain. The numerical results are similar for the case when the Neumann domain is larger than the Dirichlet one. We test the algorithm by choosing $h^0(t) = t$, $t \in (0,T]$ as an initial guess. Figure 1 gives the error reduction curves for different values of the parameter θ for $T = 2$ in (a) and $T = 200$ in (b). Note that, for a small time window, we get linear convergence for all the parameters, except for $\theta = 0.5$ which corresponds to superlinear convergence.



(a) Short time window (b) Large time window

Fig. 1 Convergence for various parameters; left: short time window, right: large time window.

For a large time window, we always observe linear convergence. We now plot the linear bound for the convergence rate in case of $\theta = 1/2$ as shown in Theorem 2. The theorem provides a T-independent theoretical bound of the error for this special relaxation parameter and this is also valid for large time windows. Eventually, a more refined analysis will give a superlinear bound shown in (9)-(10), dependent on T and the lengths of the subdomains (see [5]). Figure 2 gives a comparison between the theoretical error for the continuous model problem (calculated using inverse Laplace transforms), numerical error for the discretized problem, linear bound and the superlinear bound for $a = 2, b = 3$ and various T's. We can observe that the error curves seem to approach the linear bound as T increases.

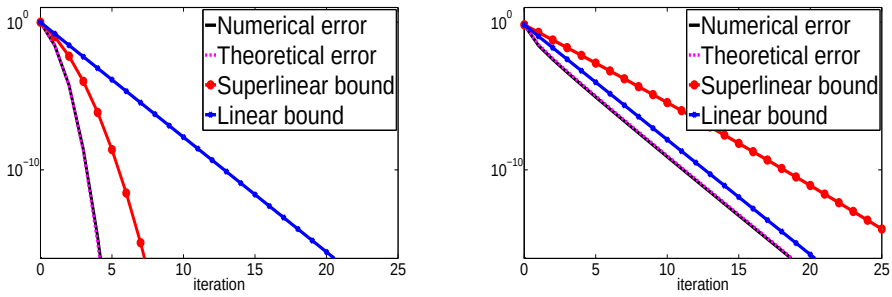


Fig. 2 Bounds for various times, $b \geq a$; in particular $a < 3b$. Left: $T = 2$, right: $T = 200$

5 Conclusions and further results

We proved convergence of the proposed DNWR algorithm in the symmetric case. For unequal subdomain lengths and for a particular choice of relaxation parameter, we presented a linear error estimate that is valid for both bounded and unbounded time intervals. In fact, Figure 2 suggests that the method converges superlinearly. To prove this, one has to consider two different cases: Dirchlet subdomain bigger than Neumann subdomain ($b \geq a$) and the other way around. Figure 3 shows $F_k(t)$ for $k = 1, 2, 3$; we see that the curves shift to the right and at the same time, the peak decreases as k increases. So, if one only considers a small time window, the peak will eventually exit the time window for k large enough and its contribution will be vanishingly small in the expression (5). This is the intuitive idea to get superlinear convergence for $\theta = 1/2$ in small time win-

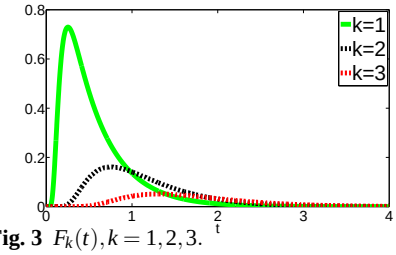


Fig. 3 $F_k(t), k = 1, 2, 3.$

dows. A detailed analysis, which is too long for this short paper, in [5] leads to the following superlinear convergence estimates for the small time window $(0,T)$:

$$\| h^k \|_{L^\infty(0,T)} \leq \left(\frac{b-a}{b}\right)^k \operatorname{erfc}\left(\frac{ka}{2\sqrt{T}}\right) \| h^0 \|_{L^\infty(0,T)}, \quad \text{for } b \geq a, \tag{9}$$

and

$$\| h^{2k} \|_{L^\infty(0,T)} \leq \left\{\frac{\sqrt{2}}{1-e^{-\frac{2k+1}{\sigma}}}\right\}^{2k} e^{-k^2/\sigma} \| h^0 \|_{L^\infty(0,T)}, \quad \text{for } b < a, \tag{10}$$

where $\sigma = T/b^2$. We are also working on a generalization of the algorithm to higher dimensions.

Acknowledgements I would like to express my gratitude to Prof. Martin J. Gander and Dr. Felix Kwok for their constant support and stimulating suggestions.

References

1. Bjørstad, P.E., Widlund, O.B.: Iterative Methods for the Solution of Elliptic Problems on Regions Partitioned into Substructures. SIAM J. Numer. Anal. (1986)
2. Bramble, J.H., Pasciak, J.E., Schatz, A.H.: An Iterative Method for Elliptic Problems on Regions Partitioned into Substructures. Mathematics of Computation (1986)
3. Churchill, R.V.: Operational Mathematics, 2nd edn. McGraw-Hill (1958)
4. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**(2), 699–732 (2006)
5. Gander, M.J., Kwok, F., Mandal, B.C.: Dirichlet-Neumann and Neumann-Neumann Waveform Relaxation Methods for the Heat Equation. (In Preparation)
6. Gander, M.J., Stuart, A.M.: Space-time continuous analysis of waveform relaxation for the heat equation. SIAM J. for Sci. Comput. **19**(6), 2014–2031 (1998)
7. Giladi, E., Keller, H.: Space time domain decomposition for parabolic problems. Numer. Math. **93**, 279–313 (2002)
8. Lelarasmee, E., Ruehli, A., Sangiovanni-Vincentelli, A.: The waveform relaxation method for time-domain analysis of large scale integrated circuits. IEEE Trans. Compt.-Aided Design Integr. Circuits Syst. **1**(3), 131–145 (1982)
9. Lindelöf, E.: Sur l'application des méthodes d'approximations successives à l'étude des intégrales réelles des équations différentielles ordinaires. Journal de Mathématiques Pures et Appliquées (1894)
10. Lions, P.L.: On the Schwarz alternating method. in First International Symposium on Domain Decomposition Methods for PDEs, I (3) (1989)
11. Martini, L., Quarteroni, A.: An iterative procedure for domain decomposition methods: a finite element approach. SIAM, in Domain Decomposition Methods for PDEs, I pp. 129–143 (1988)
12. Martini, L., Quarteroni, A.: A Relaxation Procedure for Domain Decomposition Method using Finite Elements. Numer. Math. (1989)
13. Oberhettinger, F., Badii, L.: Tables of Laplace Transforms. Springer-Verlag (1973)
14. Picard, E.: Sur l'application des méthodes d'approximations successives à l'étude de certaines équations différentielles ordinaires. Journal de Mathématiques Pures et Appliquées (1893)

Hierarchical model (Hi-Mod) reduction in non-rectilinear domains

Simona Perotto[1]

1 Introduction and motivations

In [2, 1] we have proposed an approach for the numerical modeling of second-order elliptic problems exhibiting a dominant direction in their behaviour: the solution of interest can be regarded as a main component aligned with the centerline of the domain with the addition of local perturbations along the transverse directions. Reference application is given, e.g., by advection-diffusion-reaction problems in pipes (like drug transport in the circulatory system). The basic idea of the approach is to perform a finite element discretization along the mainstream and a spectral modal approximation for the transverse components. The rationale is that the transverse components are reliably captured by few modes (usually < 10). In addition, the number of modes can locally vary along the centerline to properly fit the transverse behaviour of the solution. Thus we get an actual hierarchy of reduced models: they are essentially locally-enriched 1D models and differ for the level of detail in describing the transverse behaviour of the full problem. For this reason, we defined this approach Hierarchical Model (*Hi-Mod*) reduction.
So far we have essentially applied the Hi-Mod approach to rectilinear domains [1, 2, 4]. This implies significant simplifications in the computation of the reduced model. Nevertheless, domains with a curved centerline are clearly of paramount interest for practical applications. Aim of this paper is to perform a complete development of the Hi-Mod reduction in a generic non-rectilinear domain.

2 The geometrical setting

A Hi-Mod reduction procedure relies upon a specific shape of the computational domain $\Omega \subset \mathbb{R}^d$, with $d = 2,3$. More precisely, we assume Ω to coincide with a d-dimensional *fiber bundle*, where we distinguish a supporting one-dimensional curved domain Ω_{1D} (aligned with the mainstream), and a set of $(d-1)$-dimensional transverse fibers $\gamma \subset \mathbb{R}^{d-1}$ (associated with the transverse components of the solution). Following [1, 2], we map the current domain Ω into a reference domain, $\widehat{\Omega} = \widehat{\Omega}_{1D} \times \widehat{\gamma}_{d-1}$, with $\widehat{\Omega}_{1D}$ a straight line and $\widehat{\gamma}_{d-1}$ a reference (transverse) fiber of the same dimension as γ. For this purpose, we introduce the map $\Psi : \Omega \to \widehat{\Omega}$ and we denote by $\mathbf{z} = (x, \mathbf{y}) \in \Omega$ and $\widehat{\mathbf{z}} = (\widehat{x}, \widehat{\mathbf{y}}) \in \widehat{\Omega}$ a generic point in Ω and the corre-

[1] MOX, Dipartimento di Matematica "F. Brioschi", Politecnico di Milano, Piazza Leonardo da Vinci 32, I-20133 Milano, Italy e-mail: simona.perotto@polimi.it

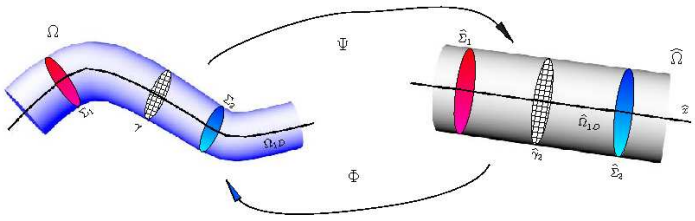


Fig. 1 Sketch of the main geometrical quantites involved in the Hi-Mod procedures ($d = 3$)

sponding point in $\widehat{\Omega}$, respectively so that $\widehat{\mathbf{z}} = \Psi(\mathbf{z}) = (\Psi_1(\mathbf{z}), \Psi_2(\mathbf{z}))$, with $\widehat{x} = \Psi_1(\mathbf{z})$ and $\widehat{\mathbf{y}} = \Psi_2(\mathbf{z})$. Likewise, we introduce the inverse map $\Phi : \widehat{\Omega} \to \Omega$, defined as $\mathbf{z} = \Phi(\widehat{\mathbf{z}}) = (\Phi_1(\widehat{\mathbf{z}}), \Phi_2(\widehat{\mathbf{z}}))$, with $x = \Phi_1(\widehat{\mathbf{z}})$ and $\mathbf{y} = \Phi_2(\widehat{\mathbf{z}})$ (see Fig. 1). Without loss of generality, we assume Ω_{1D} to coincide with the centerline of Ω, and analogously for $\widehat{\Omega}_{1D}$. We assume that both Ψ and Φ are differentiable with respect to $\mathbf{z}$. Then, we define the Jacobian associated with the map Ψ

$$\mathscr{J}(\mathbf{z}) = \frac{\partial \Psi}{\partial \mathbf{z}} = \begin{bmatrix} \dfrac{\partial \Psi_1}{\partial x} & \nabla_{\mathbf{y}} \Psi_1 \\ \dfrac{\partial \Psi_2}{\partial x} & \nabla_{\mathbf{y}} \Psi_2 \end{bmatrix} \in \mathbb{R}^{d \times d}, \tag{1}$$

where $\nabla_{\mathbf{y}}$ is the gradient with respect to $\mathbf{y}$. Notice that the first row in (1) accounts for the centerline deformation and it is not trivially the first row of the identity matrix as in the rectilinear case ([2]).

3 The Hi-Mod reduction procedure

Let us first introduce the model we aim at reducing, i.e., the so-called *full problem*. In particular, we consider directly the weak formulation, given by

$$\text{find } u \in V \quad : \quad a(u,v) = F(v) \quad \forall v \in V, \tag{2}$$

with V a Hilbert space, $a(\cdot,\cdot) : V \times V \to \mathbb{R}$ a continuous and coercive bilinear form and $F(\cdot) : V \to \mathbb{R}$ a continuous linear functional. Since we deal with second-order elliptic problems, we have $V \subseteq H^1(\Omega)$.

The Hi-Mod reduction strongly relies upon the fiber structure of Ω. The idea is to tackle the dominant and transverse components of the solution in different ways. In particular, with reference to $\widehat{\Omega}$, we introduce a one-dimensional space $V_{\widehat{\Omega}_{1D}}$ of functions compatible with the boundary conditions assigned along the extremal faces of Ω, and a modal basis $\{\varphi_k\}_{k \in \mathbb{N}^+}$ of functions orthonormal with respect to the L^2-scalar product on $\widehat{\gamma}_{d-1}$ and taking into account the boundary conditions

imposed on the lateral faces of Ω. A suitable combination of the space $V_{\widehat{\Omega}_{1D}}$ with the modal basis allows us to introduce a so-called *hierarchically reduced model*. In particular, in the following, we focus on two possible Hi-Mod reduction procedures proposed in [1, 2] and here generalized to the non-rectilinear case.

3.1 Uniform Hi-Mod reduction

The reduced space V_m characterizing a uniform Hi-Mod reduction essentially coincides with the set of the linear combinations of the modal functions whose coefficients belong to the one-dimensional space $V_{\widehat{\Omega}_{1D}}$, i.e.,

$$V_m = \Big\{ v_m(\mathbf{z}) = \sum_{k=1}^{m} v_k(\Psi_1(\mathbf{z}))\varphi_k(\Psi_2(\mathbf{z})), \text{ with } v_k \in V_{\widehat{\Omega}_{1D}} \Big\}. \tag{3}$$

The map Ψ plays a crucial role since all the functions involved are defined on the reference framework. Space V_m establishes an actual *hierarchy* of reduced models marked by the modal index m, i.e., by the different level of detail in describing the transverse behaviour of the full solution. The uniform Hi-Mod reduced formulation for (2) reads: given a modal index $m \in \mathbb{N}^+$, find $u_m \in V_m$, such that

$$a(u_m, v_m) = F(v_m) \quad \forall v_m \in V_m. \tag{4}$$

To guarantee the well-posedness and the convergence of u_m to u, we introduce a conformity ($V_m \subset V, \forall m \in \mathbb{N}^+$) and a spectral approximability ($\lim_{m\to+\infty}(\inf_{v_m\in V_m} \|v - v_m\|_V) = 0, \forall v \in V$) assumptions on V_m ([1, 2]).

Let us detail now the uniform Hi-Mod reduction procedure on a specific differential problem. In particular, we select the full model (2) as a standard linear scalar advection-diffusion-reaction (ADR) problem completed with full homogeneous Dirichlet boundary conditions, so that $V = H_0^1(\Omega)$,

$$a(u,v) = \int_\Omega \mu \nabla u \cdot \nabla v \, d\Omega + \int_\Omega \big(\mathbf{b}\cdot\nabla u + \sigma u\big) v \, d\Omega, \qquad F(v) = \int_\Omega f v \, d\Omega, \tag{5}$$

and where the following choices are made for the problem data to ensure the well-posedness of the weak form (2): $f \in L^2(\Omega)$, $\mu \in L^\infty(\Omega)$, with $\mu \geq \mu_0 > 0$ a.e. in Ω, $\sigma \in L^\infty(\Omega)$, $\mathbf{b} = (b_1, \mathbf{b}_2)^T \in L^\infty(\Omega) \times [L^\infty(\Omega)]^{d-1}$, with $\nabla\cdot\mathbf{b} \in L^\infty(\Omega)$ and such that $-\frac{1}{2}\nabla\cdot\mathbf{b} + \sigma \geq 0$ a.e. in Ω.
Now we consider the reduced model (4); we replace u_m with the corresponding modal representation $u_m(\mathbf{z}) = \sum_{j=1}^m u_j(\Psi_1(\mathbf{z}))\varphi_j(\Psi_2(\mathbf{z}))$ and v_m with the product $\vartheta(\Psi_1(\mathbf{z}))\varphi_k(\Psi_2(\mathbf{z}))$, where $\vartheta, u_j \in V_{\widehat{\Omega}_{1D}} = H_0^1(\widehat{\Omega}_{1D})$ for $j = 1, \ldots, m$, to get

$$\sum_{j=1}^{m} \Big[\int_\Omega \mu(\mathbf{z}) \nabla\big(u_j(\Psi_1(\mathbf{z}))\,\varphi_j(\Psi_2(\mathbf{z}))\big) \cdot \nabla\big(\vartheta(\Psi_1(\mathbf{z}))\,\varphi_k(\Psi_2(\mathbf{z}))\big)\, d\Omega \tag{6}$$

$$
\begin{aligned}
&+\int_\Omega \mathbf{b}(\mathbf{z})\cdot\nabla\big(u_j(\Psi_1(\mathbf{z}))\,\varphi_j(\Psi_2(\mathbf{z}))\big)\,\vartheta(\Psi_1(\mathbf{z}))\,\varphi_k(\Psi_2(\mathbf{z}))\,d\Omega\\
&+\int_\Omega \sigma(\mathbf{z})u_j(\Psi_1(\mathbf{z}))\,\varphi_j(\Psi_2(\mathbf{z}))\vartheta(\Psi_1(\mathbf{z}))\,\varphi_k(\Psi_2(\mathbf{z}))\,d\Omega\Big]\\
&=\int_\Omega f(\mathbf{z})\vartheta(\Psi_1(\mathbf{z}))\,\varphi_k(\Psi_2(\mathbf{z}))\,d\Omega,
\end{aligned}
$$

where ∇ denotes the gradient with respect to $\mathbf{z}$. The actual unknowns of the Hi-Mod reduced formulation (4) are the modal coefficients $u_j \in V_{\widehat{\Omega}_{1D}}$. We expand separately the four integrals, by exploiting the gradient expansion

$$
\begin{aligned}
&\nabla(w(\Psi_1(\mathbf{z}))\varphi_s(\Psi_2(\mathbf{z})))=\\
&w'(\Psi_1(\mathbf{z}))\varphi_s(\Psi_2(\mathbf{z}))\begin{bmatrix}\dfrac{\partial\Psi_1(\mathbf{z})}{\partial x}\\ \nabla_{\mathbf{y}}\Psi_1(\mathbf{z})\end{bmatrix}+w(\Psi_1(\mathbf{z}))\varphi_s'(\Psi_2(\mathbf{z}))\begin{bmatrix}\dfrac{\partial\Psi_2(\mathbf{z})}{\partial x}\\ \nabla_{\mathbf{y}}\Psi_2(\mathbf{z})\end{bmatrix},
\end{aligned}
$$

where $w'(\Psi_1(\mathbf{z}))=dw/d\widehat{x}|_{\widehat{x}=\Psi_1(\mathbf{z})}$, $\varphi_s'(\Psi_2(\mathbf{z}))=d\varphi_s/d\widehat{\mathbf{y}}|_{\widehat{\mathbf{y}}=\Psi_2(\mathbf{z})}$ and with $w\in V_{\widehat{\Omega}_{1D}}$. The idea is to rewrite each term on the reference domain by properly exploiting the maps Ψ, Φ. Let us first consider the diffusive contribution in (6):

$$
\begin{aligned}
&\int_{\widehat{\Omega}}\ \mu(\Phi(\widehat{\mathbf{z}}))\Big\{\Big[\Big(\frac{\partial\Psi_1(\Phi(\widehat{\mathbf{z}}))}{\partial x}\Big)^2+\big(\nabla_{\mathbf{y}}\Psi_1(\Phi(\widehat{\mathbf{z}}))\big)^2\Big]\varphi_j(\widehat{\mathbf{y}})\varphi_k(\widehat{\mathbf{y}})u_j'(\widehat{x})\vartheta'(\widehat{x})\\
&+\ \Big[\frac{\partial\Psi_1(\Phi(\widehat{\mathbf{z}}))}{\partial x}\frac{\partial\Psi_2(\Phi(\widehat{\mathbf{z}}))}{\partial x}+\nabla_{\mathbf{y}}\Psi_1(\Phi(\widehat{\mathbf{z}}))\nabla_{\mathbf{y}}\Psi_2(\Phi(\widehat{\mathbf{z}}))\Big]\\
&\quad\big[\varphi_j(\widehat{\mathbf{y}})\varphi_k'(\widehat{\mathbf{y}})u_j'(\widehat{x})\vartheta(\widehat{x})+\varphi_j'(\widehat{\mathbf{y}})\varphi_k(\widehat{\mathbf{y}})u_j(\widehat{x})\vartheta'(\widehat{x})\big]\\
&+\ \Big[\Big(\frac{\partial\Psi_2(\Phi(\widehat{\mathbf{z}}))}{\partial x}\Big)^2+\big(\nabla_{\mathbf{y}}\Psi_2(\Phi(\widehat{\mathbf{z}}))\big)^2\Big]\varphi_j'(\widehat{\mathbf{y}})\varphi_k'(\widehat{\mathbf{y}})u_j(\widehat{x})\vartheta(\widehat{x})\Big\}\,|\mathscr{I}^{-1}(\Phi(\widehat{\mathbf{z}}))|\,d\widehat{\Omega},
\end{aligned}
\tag{7}
$$

with $\mathscr{I}$ the Jacobian defined in (1). The convective term is changed into

$$
\begin{aligned}
\int_{\widehat{\Omega}}\Big\{\ &\Big[b_1(\Phi(\widehat{\mathbf{z}}))\frac{\partial\Psi_1(\Phi(\widehat{\mathbf{z}}))}{\partial x}+\mathbf{b}_2(\Phi(\widehat{\mathbf{z}}))\nabla_{\mathbf{y}}\Psi_1(\Phi(\widehat{\mathbf{z}}))\Big]\varphi_j(\widehat{\mathbf{y}})\varphi_k(\widehat{\mathbf{y}})u_j'(\widehat{x})\vartheta(\widehat{x})\\
&\Big[b_1(\Phi(\widehat{\mathbf{z}}))\frac{\partial\Psi_2(\Phi(\widehat{\mathbf{z}}))}{\partial x}+\mathbf{b}_2(\Phi(\widehat{\mathbf{z}}))\nabla_{\mathbf{y}}\Psi_2(\Phi(\widehat{\mathbf{z}}))\Big]\varphi_j'(\widehat{\mathbf{y}})\varphi_k(\widehat{\mathbf{y}})u_j(\widehat{x})\vartheta(\widehat{x})\Big\}\\
&|\mathscr{I}^{-1}(\Phi(\widehat{\mathbf{z}}))|\,d\widehat{\Omega},
\end{aligned}
\tag{8}
$$

while, for the reactive term, we have

$$
\int_{\widehat{\Omega}}\sigma(\Phi(\widehat{\mathbf{z}}))\varphi_j(\widehat{\mathbf{y}})\varphi_k(\widehat{\mathbf{y}})u_j(\widehat{x})\vartheta(\widehat{x})|\mathscr{I}^{-1}(\Phi(\widehat{\mathbf{z}}))|\,d\widehat{\Omega}. \tag{9}
$$

Finally, for the source term in (6), we simply obtain

$$
\int_{\widehat{\Omega}} f(\Phi(\widehat{\mathbf{z}}))\varphi_k(\widehat{\mathbf{y}})\vartheta(\widehat{x})|\mathscr{I}^{-1}(\Phi(\widehat{\mathbf{z}}))|\,d\widehat{\Omega}. \tag{10}
$$

From (7) we notice that the treatment of the diffusive term generates advective and reactive contributions in the reduced setting. Similarly, the reduced convection term (8) features also a reactive contribution. A straightforward combination of (7)-(10) leads to the following Hi-Mod reduced formulation for the ADR problem defined in (5): find $u_j \in V_{\widehat{\Omega}_{1D}}$ with $j = 1,\dots,m$, such that, for any $\vartheta \in V_{\widehat{\Omega}_{1D}}$ and $k = 1,\dots,m$,

$$\sum_{j=1}^{m} \Big\{ \int_{\widehat{\Omega}_{1D}} \Big[\widehat{r}_{kj}^{1,1}(\widehat{x})\, u_j'(\widehat{x})\, \vartheta'(\widehat{x}) + \widehat{r}_{kj}^{1,0}(\widehat{x})\, u_j'(\widehat{x})\, \vartheta(\widehat{x}) + \widehat{r}_{kj}^{0,1}(\widehat{x})\, u_j(\widehat{x})\, \vartheta'(\widehat{x}) \qquad (11)$$
$$+ \ \widehat{r}_{kj}^{0,0}(\widehat{x})\, u_j(\widehat{x})\, \vartheta(\widehat{x}) \Big]\, d\widehat{x} \Big\} = \int_{\widehat{\Omega}_{1D}} \Big[\int_{\widehat{\gamma}_{d-1}} f(\Phi(\widehat{\mathbf{z}}))\varphi_k(\widehat{\mathbf{y}}) |\mathscr{J}^{-1}(\Phi(\widehat{\mathbf{z}}))|\, d\widehat{\mathbf{y}} \Big] \vartheta(\widehat{x})\, d\widehat{x},$$

where

$$\widehat{r}_{kj}^{s,t}(\widehat{x}) = \int_{\widehat{\gamma}_{d-1}} r_{kj}^{s,t}(\widehat{x},\widehat{\mathbf{y}})\, |\mathscr{J}^{-1}(\Phi(\widehat{\mathbf{z}}))|\, d\widehat{\mathbf{y}}, \quad s,t = 0,1, \quad k = 1,\dots,m, \qquad (12)$$

with

$$\begin{aligned}
r_{kj}^{1,1}(\widehat{\mathbf{z}}) &= \mu(\Phi(\widehat{\mathbf{z}}))\, \alpha_1(\widehat{\mathbf{z}})\, \varphi_j(\widehat{\mathbf{y}})\varphi_k(\widehat{\mathbf{y}}), \quad r_{kj}^{0,1}(\widehat{\mathbf{z}}) = \mu(\Phi(\widehat{\mathbf{z}}))\, \delta(\widehat{\mathbf{z}})\, \varphi_j'(\widehat{\mathbf{y}})\varphi_k(\widehat{\mathbf{y}}), \\
r_{kj}^{1,0}(\widehat{\mathbf{z}}) &= \mu(\Phi(\widehat{\mathbf{z}}))\, \delta(\widehat{\mathbf{z}})\, \varphi_j(\widehat{\mathbf{y}})\varphi_k'(\widehat{\mathbf{y}}) + \beta_1(\widehat{\mathbf{z}})\, \varphi_j(\widehat{\mathbf{y}})\varphi_k(\widehat{\mathbf{y}}), \qquad (13) \\
r_{kj}^{0,0}(\widehat{\mathbf{z}}) &= \mu(\Phi(\widehat{\mathbf{z}}))\, \alpha_2(\widehat{\mathbf{z}})\varphi_j'(\widehat{\mathbf{y}})\varphi_k'(\widehat{\mathbf{y}}) + \beta_2(\widehat{\mathbf{z}})\, \varphi_j'(\widehat{\mathbf{y}})\varphi_k(\widehat{\mathbf{y}}) + \sigma(\Phi(\widehat{\mathbf{z}}))\, \varphi_j(\widehat{\mathbf{y}})\varphi_k(\widehat{\mathbf{y}}),
\end{aligned}$$

and

$$\begin{aligned}
\alpha_i(\widehat{\mathbf{z}}) &= \Big(\frac{\partial \Psi_i(\Phi(\widehat{\mathbf{z}}))}{\partial x} \Big)^2 + \big(\nabla_{\mathbf{y}} \Psi_i(\Phi(\widehat{\mathbf{z}})) \big)^2 \quad i = 1,2, \\
\beta_i(\widehat{\mathbf{z}}) &= b_1(\Phi(\widehat{\mathbf{z}})) \frac{\partial \Psi_i(\Phi(\widehat{\mathbf{z}}))}{\partial x} + \mathbf{b}_2(\Phi(\widehat{\mathbf{z}})) \cdot \nabla_{\mathbf{y}} \Psi_i(\Phi(\widehat{\mathbf{z}})) \quad i = 1,2, \qquad (14) \\
\delta(\widehat{\mathbf{z}}) &= \frac{\partial \Psi_1(\Phi(\widehat{\mathbf{z}}))}{\partial x} \frac{\partial \Psi_2(\Phi(\widehat{\mathbf{z}}))}{\partial x} + \nabla_{\mathbf{y}} \Psi_1(\Phi(\widehat{\mathbf{z}})) \cdot \nabla_{\mathbf{y}} \Psi_2(\Phi(\widehat{\mathbf{z}})).
\end{aligned}$$

In the reduced model (11) the dependence of the solution on the dominant and on the transverse directions is split. The Hi-Mod reduction procedure yields a *special one-dimensional model* associated with the main curved stream, whose coefficients, $\widehat{r}_{kj}^{s,t}$, are properly enriched to include the effects of the transverse components. In particular, the coefficients in (13) reduce to the ones in [1] for rectilinear domains, where $\partial \Psi_1 / \partial x = 1$ and $\nabla_{\mathbf{y}} \Psi_1 = 0$. From a computational viewpoint, the solution to (11) requires solving a system of m coupled one-dimensional problems instead of a full d-dimensional problem. Following [1, 2], we discretize these 1D problems by introducing a finite element discretization along $\widehat{\Omega}_{1D}$, while preserving the modal expansion in correspondence with the transverse directions. We are led to solve a linear system with an $m \times m$ block matrix, where each block is an $N_h \times N_h$ matrix with the sparsity pattern of the selected finite element space X_h, with $\dim(X_h) = N_h$. An appropriate choice of the modal index m in (3) is certainly a critical issue of the uniform Hi-Mod reduction. In [2] a "trial and error" approach is suggested:

we move from the computationally cheapest choice $m = 1$ and then we gradually increase such a value until the addition of the successive modal function does not significantly improve the accuracy of the reduced solution. This strategy may be sometimes speeded up, e.g., when a partial physical knowledge of the phenomenon at hand is available, so that the initial guess can be properly calibrated.

3.2 Piecewise Hi-Mod reduction

The uniform approach may become really uneffective when the meaningful transverse components of the solution are strongly localized: a large number of modal functions is employed on the whole Ω, even though it would be strictly necessary only where significant transverse components are present. This justifies the proposal of a new formulation, where a different number of modes is employed in different parts of Ω: many modes where the transverse components are important, few modes where these are less significant. The modal index m becomes therefore a piecewise constant vector: this justifies the name of this approach. In more detail, let us assume to locate s subdomains Ω_i in Ω such that $\overline{\Omega} = \cup_{i=1}^{s} \overline{\Omega}_i$, with $\Sigma_i = \overline{\Omega}_i \cap \overline{\Omega}_{i+1}$ the interface between Ω_i and Ω_{i+1}, and let $\{\widehat{\Omega}_i\}_{i=1}^{s}$ be the corresponding partition on $\widehat{\Omega}$, with $\widehat{\Sigma}_i = \Psi(\Sigma_i) = \overline{\widehat{\Omega}}_i \cap \overline{\widehat{\Omega}}_{i+1}$ (see Fig. 1). In particular, we employ m_i modal functions on Ω_i, for $i = 1, \ldots, s$. Following [3], the piecewise Hi-Mod reduced formulation for (2) reads: given a modal multi-index $\mathbf{m} = \{m_i\}_{i=1}^{s} \in [\mathbb{N}^+]^s$, find $u_{\mathbf{m}} \in V_{\mathbf{m}}^b$, such that

$$a_\Omega(u_{\mathbf{m}}, v_{\mathbf{m}}) = F_\Omega(v_{\mathbf{m}}) \quad \forall v_{\mathbf{m}} \in V_{\mathbf{m}}^b, \tag{15}$$

where $a_\Omega(u_{\mathbf{m}}, v_{\mathbf{m}}) = \sum_{i=1}^{s} a_i(u_{\mathbf{m}}|_{\Omega_i}, v_{\mathbf{m}}|_{\Omega_i})$, $F_\Omega(v_{\mathbf{m}}) = \sum_{i=1}^{s} F_i(v_{\mathbf{m}}|_{\Omega_i})$ with $a_i(\cdot,\cdot)$ and $\mathscr{F}_i(\cdot)$ the restriction to Ω_i of the bilinear and of the linear form in (2), respectively. The reduced space in (15) is a subset of the broken Sobolev space $H^1(\Omega, \mathscr{T}_\Omega)$ associated with the partition $\mathscr{T}_\Omega = \{\Omega_i\}_{i=1}^{s}$, and it is defined by

$$\begin{aligned} V_{\mathbf{m}}^b = \Big\{ & v_{\mathbf{m}} \in L^2(\Omega) \ : \ v_{\mathbf{m}}|_{\Omega_i}(\mathbf{z}) = \sum_{k=1}^{m_i} v_k^i(\Psi_1(\mathbf{z}))\, \varphi_k(\Psi_2(\mathbf{z})) \in H^1(\Omega_i) \\ & \forall i = 1, \ldots, s, \text{ with } v_k^i \in H^1(\widehat{\Omega}_{1D,i}) \text{ and s.t., } \forall k = 1, \ldots, m_\perp^j \text{ with } j = 1, \ldots, s-1, \\ & \int_{\widehat{\gamma}_{d-1}} \left[v_{\mathbf{m}}|_{\Omega_{j+1}}(\Phi(\widehat{\Sigma}_j)) - v_{\mathbf{m}}|_{\Omega_j}(\Phi(\widehat{\Sigma}_j)) \right] \varphi_k(\widehat{\mathbf{y}})\, d\widehat{\mathbf{y}} = 0 \Big\}, \end{aligned}$$

with $m_\perp^j = \min(m_j, m_{j+1})$ and $\widehat{\Omega}_{1D,i} = \widehat{\Omega}_{1D} \cap \widehat{\Omega}_i$. The integral condition weakly enforces the continuity of the solution in correspondence with the minimum number of modes employed on the whole Ω. This does not guarantee *a priori* the conformity of the reduced solution $u_{\mathbf{m}}$ (see section 4.2.2 in [2] for more details). According to [3], we resort to a relaxed iterative substructuring Dirichlet/Neumann method to impose the weak continuity at the interfaces. From a computational viewpoint, at each iteration of the Dirichlet/Neumann scheme, we apply a uniform Hi-Mod reduction on each subdomain Ω_i, i.e., we solve s systems of coupled 1D problems which are

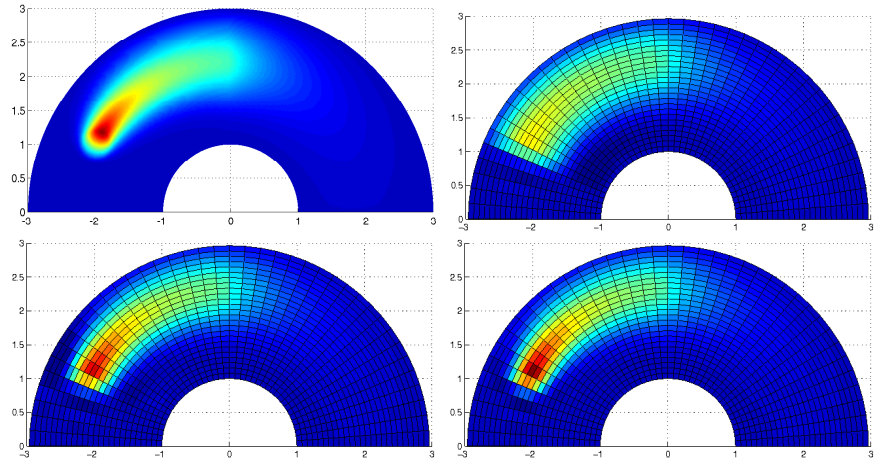


Fig. 2 Full solution and uniform Hi-Mod reduced solutions u_3, u_5, u_7 (top-bottom, left-right)

suitably approximated via a finite element discretization along $\widehat{\Omega}_{1D}$, analogously to the uniform case. The choice of the modal multi-index $\mathbf{m}$ in (15) is hereafter based on an *a priori* approach, driven by some knowledge of the solution u. The generalization of the approach proposed in [3] for rectilinear domains, where an *a posteriori* modeling error estimator drives the automatic selection of both the Ω_i's and $\mathbf{m}$ is a possible follow up of this work.

4 Numerical results

We numerically assess the two proposed Hi-Mod reduction procedures in a two-dimensional setting. In particular, we use affine finite elements to discretize the problem along $\widehat{\Omega}_{1D}$, while employing sinusoidal functions to model the transverse components. We evaluate the integrals of the sine functions via Gaussian quadrature formulas, with, at least, four quadrature nodes per wavelenght. Of course, different choices are possible for the modal basis (Legendre polynomials, wavelets, suitable eigenfunctions).

We reduce the ADR problem defined in (5) on the annular region Ω between the two concentric circles $x^2+y^2=1$ and $x^2+y^2=9$. We select $\mu=1$, the circular clockwise advective field $\mathbf{b}=\big(30\sin(\text{atan2}(y,x)),-30\cos(\text{atan2}(y,x))\big)^T$, with $-\pi\le\text{atan2}(y,x)\le\pi$, $\sigma=30\chi_+$ with $\chi_+=\{(x,y)\in\Omega : x>0\}$, and the source term $f=1000\chi_D$ localized in the small circular region $D=\{(x,y):(x+2)^2+(y-1)^2<0.05\}$. Finally, full homogeneous Dirichlet boundary conditions complete the problem. The choice of the data identifies a full solution characterized by a peak in D; it is convected by the field $\mathbf{b}$ and damped by the reaction (see Fig. 2, top-left).
Figure 2 gathers the reduced solutions provided by the uniform Hi-Mod reduction

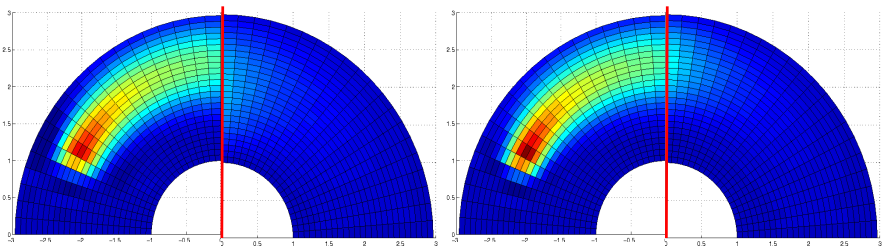


Fig. 3 Piecewise Hi-Mod reduced solutions $u_{\{5,1\}}$ (left) and $u_{\{7,3\}}$ (right)

for different choices of the modal index m and when a uniform finite element discretization of size $h = \pi/40$ is employed on $\widehat{\Omega}_{1D}$. Solution u_3 clearly fails in detecting the peak in D. At least seven modal functions are demanded to get a reliable reduced model: the peak of u is well captured for this choice, while the successive modes essentially do not improve the accuracy of u_m.

The most significant localization of the transverse components in the left part of Ω suggests us employing a higher number of modes in this part of the domain, according to a piecewise Hi-Mod reduction. We split Ω into two subdomains via the interface $\Sigma_1 = \{0\} \times (1,3)$; then we make two different choices for the modal multi-index, $\mathbf{m} = \{5,1\}$ and $\mathbf{m} = \{7,3\}$, while preserving the finite element partition of the uniform approach. Concerning the domain decomposition algorithm, we set the convergence tolerance for the relative error to 10^{-3} and the relaxation parameter to 0.5. Moreover, to guarantee the well-posedness of the ADR subproblems, we assign the Dirichlet and the Neumann condition on the right- and on the left-hand side of Σ_1, respectively. The algorithm converges after ten iterations for both choices of $\mathbf{m}$. Figure 3 shows the reduced solutions $u_{\{5,1\}}$ (left) and $u_{\{7,3\}}$ (right) at the last iteration. As expected, $u_{\{7,3\}}$ provides a better approximation of the full solution; in particular, by comparing the color maps, we can state that $u_{\{7,3\}}$ essentially coincides with u_7 in Fig. 2, bottom-right. Finally, according to [2], both $u_{\{5,1\}}$ and $u_{\{7,3\}}$ are H^1-conforming approximations: the model discontinuity across Σ_1 is therefore consequence of the truncation of the iterative domain decomposition algorithm.

References

1. Ern, A., Perotto, S., Veneziani, A.: Hierarchical model reduction for advection-diffusion-reaction problems. In: K. Kunisch, G. Of, O. Steinbach (eds.) Numerical Mathematics and Advanced Applications, pp. 703–710. Springer–Verlag (2008)
2. Perotto, S., Ern, A., Veneziani, A.: Hierarchical local model reduction for elliptic problems: a domain decomposition approach. Multiscale Modeling & Simulation **8**(4), 1102–1127 (2010)
3. Perotto, S., Veneziani, A.: Coupled model and grid adaptivity in hierarchical reduction of elliptic problems.. J. Sci. Comput. (2014). DOI:10.1007/s10915-013-9804-y
4. Perotto, S., Zilio, A.: Hierarchical model reduction: three different approaches. In: A. Cangiani, R. Davidchack, E. Georgoulis, A. Gorban, J. Levesley, M. Tretyakov (eds.) Numerical Mathematics and Advanced Applications. Springer–Verlag (2013)

The Origins of the Alternating Schwarz Method

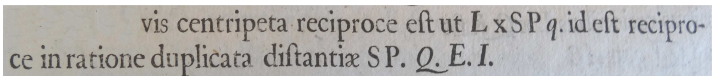
Martin J. Gander[1] and Gerhard Wanner[1]

1 Introduction

Schwarz methods are nowadays known as parallel solvers, and there are many variants: alternating and parallel Schwarz methods at the continuous level, additive and multiplicative Schwarz methods at the discrete level, also with restricted variants, which in the additive case build the important bridge between discrete and continuous Schwarz methods, see [4]. But where did these methods come from? Why were they invented in the first place? We explain in this paper that Hermann Amandus Schwarz invented the alternating Schwarz method in [18] to close an important gap in the proof of the Riemann mapping theorem, which was based on the Dirichlet principle. The Dirichlet principle itself addresses the important question of existence and uniqueness of solutions of Laplace's equation on a bounded domain with Dirichlet boundary conditions, and in the 19th century, this equation appeared independently in many different areas. It was therefore of fundamental importance to put the Dirichlet principle on firm mathematical grounds, and this is one of the major achievements of Schwarz.

2 Laplace's equation

In his Principia in 1687, Newton presented among many results also his famous inverse square law for celestial bodies [15, end of proof of Prop. XI] [1]:


vis centripeta reciproce eſt ut L x S P q. id eſt recipro-
ce in ratione duplicata diſtantiæ S P. Q. E. I.

see also [20] for a comprehensive treatment of the influence of Kepler and Newton on numerical analysis. In modern notation, if we denote by $\mathbf{f}$ the force between two celestial bodies, then $\mathbf{f}$ is proportional to $\frac{1}{r^2}$, where $r := \sqrt{(x-\xi)^2+(y-\eta)^2+(z-\zeta)^2}$, using the notation in Figure 1. Writing $\mathbf{f} = (f_1, f_2, f_3)$ component-wise, we obtain for the components

$$f_1 \approx \frac{x-\xi}{r^3}, \quad f_2 \approx \frac{y-\eta}{r^3}, \quad f_1 \approx \frac{z-\zeta}{r^3}.$$

[1]University of Geneva, Section of Mathematics, 2-4 rue du Lièvre, CP 64, CH-1211 Geneva 4, e-mail: {Martin.Gander}{Gerhard.Wanner}@unige.ch

[1] The centripetal force is inverse to $L \times SP^2$, it is inversely proportional to the squared distance SP. Q.E.I.

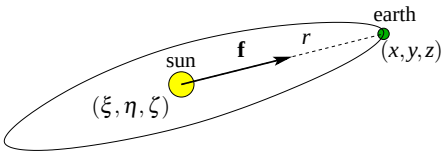


Fig. 1 The sun and our planet earth, for which Newton's inverse square law holds

This very elegant and simple law is at first only valid for point masses. Laplace then, from 1785 onwards, was wondering how these forces look like if the body is not a point, but a three dimensional irregular object occupying a domain $V \subset \mathbb{R}^3$. A clear exposition of his ideas only appeared in his *Mécanique Céleste* from 1799, see [9]. He imagined that the body is composed of molecules, see the original reproduced in Figure 2. In that case, one would need to sum the contributions of all the infinitesimally small body parts ("molecules") making up the entire volume, and would thus obtain for example for the first component of the force

$$f_1 = \int_V \rho(\xi,\eta,\zeta)\frac{x-\xi}{r^3}\,d\xi\,d\eta\,d\zeta, \tag{1}$$

where ρ denotes the density of the body. The key idea of Laplace was now to introduce the potential function

$$u = \int\int\int \rho(\xi,\eta,\zeta)\frac{1}{r}d\xi d\eta d\zeta. \tag{2}$$

11. Soient x, y, z, les trois coordonnées du point attiré que nous désignerons par m; soit dM une molécule du sphéroïde, et x', y', z', les coordonnées de cette molécule; si l'on nomme ρ sa densité, ρ étant une fonction de x', y', z', indépendante de x, y, z; on aura

$$dM = \rho.dx'.dy'.dz',$$

L'action de dM sur m, décomposée parallèlement à l'axe des x, et dirigée vers leur origine, sera

$$\frac{\rho.dx'.dy'.dz'.(x-x')}{\{(x-x')^2+(y-y')^2+(z-z')^2\}^{\frac{3}{2}}};$$

et par conséquent elle sera égale à

$$-\left\{d.\frac{\rho.dx'.dy'.dz'}{\sqrt{(x-x')^2+(y-y')^2+(z-z')^2}}\right\}:dx;$$

en nommant donc V, l'intégrale

$$\int\frac{\rho.dx'.dy'.dz'}{\sqrt{(x-x')^2+(y-y')^2+(z-z')^2}},$$

étendue à la masse entière du sphéroïde; on aura $-\left(\frac{dV}{dx}\right)$, pour l'action totale du sphéroïde sur le point m, décomposée parallèlement à l'axe des x, et dirigée vers leur origine.

Let x, y, z, be the coordinates of the attracted point m; let dM be a molecule of a spherical body with coordinates x', y', z'; if we call ρ the density, function of x', y', z', independent of x, y, z, we get

The action of dM on m, decomposed parallel to the x-axis, and directed towards its origin, is

and hence it will be equal to

denoting by V the integral

extended to the entire mass of the spherical body, we will have $-(\frac{dV}{dx})$, for the total action of the spherical body on the point m, decomposed in parallel to the x-axis and directed towards their origin.

Fig. 2 Generalization of Laplace of the inverse square law of Newton to the case of a spherical body, arguing with molecules. Copied from the 1799 publication of Laplace's *Mécanique Céleste* [9, page 136].

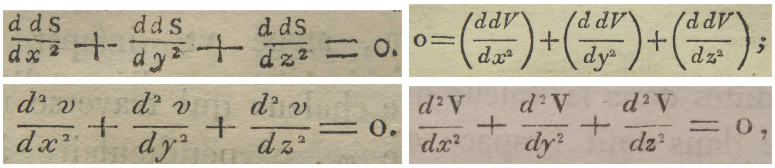


Fig. 3 Laplace equation by Euler in 1752 (top left), by Laplace in 1799 (top right), by Fourier in 1822 (bottom left), and by Kelvin in 1847 (bottom right)

Taking a derivative with respect x, and using $\frac{\partial}{\partial x}\frac{1}{r} = -\frac{x-\xi}{r^3}$, we obtain by comparing with (1), after a similar computation for y and z,

$$\mathbf{f} = -\left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}\right). \tag{3}$$

Differentiating once more, we obtain $\frac{\partial}{\partial x}\frac{x-\xi}{r^3} = \frac{r^3 - 3(x-\xi)^2 r}{r^6}$, and therefore, performing the same steps for y and z as well, that the potential function satisfies

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 0, \quad \text{Laplace's equation!} \tag{4}$$

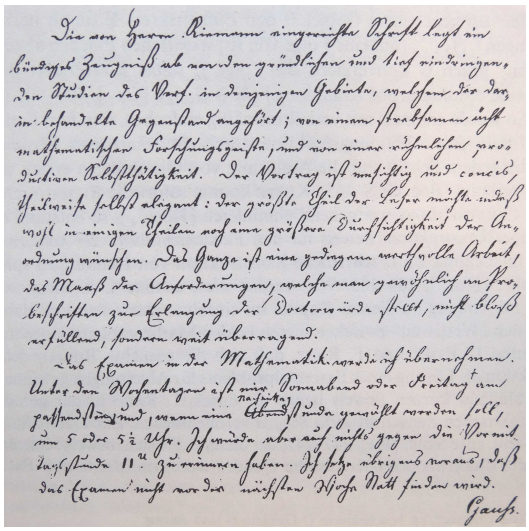
This equation appeared already in Euler's *Principia motus fluidorum* [2] (E258, written 1752, published 1756) see Figure 3, but Euler could not really use it. It appeared again in the theory of heat transfer, published by Fourier [3] in 1822, see Figure 3. Fourier also argued with molecules, and Newton's law of cooling, in order to derive the equation.

Laplace's equation turned out to be absolutely fundamental, it appeared again in the theory of magnetism proposed by Gauss and Weber in Göttingen in 1839, in the theory of electric fields put forward by W. Thomson (the later Lord Kelvin, published in the Liouville Journal from 1847 on pages 256 and 496), in conformal maps (Gauss 1825), in the irrotational motion of fluids in two dimensions (Helmholtz 1858), and finally in complex analysis, in particular in Riemann's PhD Thesis in 1851, which is available in a modern typeset version in [17].

3 The Riemann Mapping Theorem

Riemann was a prodigy already in high-school, and his mathematical talent impressed everybody:

> "Ein Lehrer, der Rektor Schmalfuss, lieh ihm Legendres Zahlentheorie (Théorie des Nombres), ein schwieriges Werk von 859 Quartformat-Seiten, bekam sie aber schon eine Woche



"Die von Herrn Riemann eingereichte Schrift legt ein bündiges Zeugniss ab von den gründlichen und tief eindringenden Studien des Verf. in demjenigen Gebiete, welchem der darin behandelte Gegenstand angeört; von einem strebsamen ächt mathematischen Forschungsgeiste, und von einer rühmlichen productiven Selbstthätigkeit. Der Vortrag ist umsichtig und concis, theilweise selbst elegant: der grösste Theil der Leser möchte indess wohl in einigen Theilen noch eine grössere Durchsichtigkeit der Anordung wünschen. Das Ganze ist eine gediegene werthvolle Arbeit, das Maass der Anforderungen, welche man gewöhnlich an Probeschriften zur Erlangung der Doctorwürde stellt, nicht bloss erfüllend, sondern weit überragend .

Das Examen in der Mathematik werde ich übernehmen. Unter den Wochentagen ist mir Sonnabend oder Freitag am passendsten und, wenn eine Nachmittagsstunde gewählt werden soll, um 5 oder 5 1/2 Uhr. Ich würde aber auch nichts gegen die Vormittagsstunde 11h zu erinnern haben. Ich setze übrigens voraus, dass das Examen nicht vor der nächsten Woche statt finden wird".

Fig. 4 Handwritten Laudatio of Gauss on Riemann's PhD thesis, copied from Remmert [16]

> später zurück und fand, als er Riemann im Abitur über dieses Werk weit über das Übliche hinaus prüfte, dass Riemann sich dieses Buch vollständig zu eigen gemacht hatte." [2]

Riemann's PhD supervisor was Gauss, who rarely praised the work of other mathematicians. We show the laudatio on Riemann's thesis in the original handwriting of Gauss in Figure 4 [3]. Riemann build in his thesis the foundation of analytic function theory, and gave toward the end an example, which became the famous Riemann Mapping theorem:

---

[2] "A teacher, Professor Schmalfuss, lend him Legendre's book on number theory, a very difficult work of 859 pages in quarto format, and he got it back already after a week. When he tested Riemann in his final high-school exam on this subject much more thoroughly than usual, he realized that Riemann had completely mastered the content of the book."

[3] The manuscript submitted by Riemann is a testament of the thorough and deep studies by the author in the area to which the treated subject belongs; of an aspiring and truly mathematical research spirit, and of a glorious, productive self-activity. The presentation is comprehensive and concise, partly even elegant: the major part of the readers would however in some parts still wish for more transparency and better arrangement. As a whole, it is a dignified valuable work, which does not only satisfy the requirement one usually imposes on a manuscript to obtain a PhD degree, but goes very far beyond.
The mathematics exam I will do myself. I prefer Sunday or Friday, and in the afternoon at 5 or 5:30 pm. I would also be available in the morning at 11am. I assume that the exam will not be before next week.

> "Zwei gegebene einfach zusammenhängende Flächen können stets so aufeinander bezogen werden, dass jedem Punkte der einen ein mit ihm stetig fortrückender Punkt entspricht...;"[4]

Riemann also gave a constructive proof of this theorem. In modern notation, we need to find an analytic function f which maps Ω to the unit disk and one point $z_0 \in \Omega$ into 0. We thus set $f(z) := (z-z_0)e^{g(z)}$, $g = u+iv$ an analytic function to be determined, in order to ensure that z_0 is the only point mapped into zero. In order to arrive from the boundary $\partial\Omega$ on the boundary of the disk with the mapping, we must have for all $z \in \partial\Omega$ that $|f(z)| = 1$, which implies that

$$1 = |f(z)| = |(z-z_0)e^{u+iv}| = |(z-z_0)|e^u \Longrightarrow\ u(z) = -log|z-z_0|,\ \forall z \in \partial\Omega. \quad (5)$$

Since g is analytic, the real part u of g satisfies Laplace's equation $\Delta u = 0$ on Ω, with boundary values given in (5). It thus suffices to solve for u, construct v using the Cauchy-Riemann equations, and then the construction of f is complete.

Riemann's PhD thesis was very well received by the mathematical world of that time, and widely studied. Among the first readers were also Weierstrass and Helmholtz:

> "Weierstrass hatte die Riemannsche Dissertation zum Ferienstudium mitgenommen und klagte, dass ihm, dem Funktionentheoretiker, die Riemannschen Methoden schwer verständlich seien. Helmholtz bat sich die Schrift aus und sagte beim nächsten Zusammentreffen, ihm schienen die Riemannschen Gedankengänge völlig naturgemäss und selbstverständlich zu sein." (Funktionentheorie 1 von Reinhold Remmert, Georg Schumacher) [5]

Nevertheless, an important question remained: Riemann had used that a u satisfying Laplace's equation on an arbitrary domain with given boundary conditions exists. But was this really true ? When Riemann was challenged with this, he replied

> "Hierzu kann in vielen Fällen ... ein Princip dienen, welches Dirichlet zur Lösung dieser Aufgabe für eine der Laplace'schen Differentialgleichung genügende Function ... in seinen Vorlesungen ... seit einer Reihe von Jahren zu geben pflegt." (Riemann 1857, *Werke* p. 97)[6]

The idea, which became known under the name of "Dirichlet principle", is to choose among all the functions defined on a given domain Ω with the prescribed boundary values the one that minimizes the integral

$$J(u) = \iint_\Omega \frac{1}{2}\left(u_x^2 + u_y^2\right) dx\,dy \quad \text{which is always non-negative.}$$

But is the Dirichlet principle correct for an arbitrary, non-negative functional? Weierstrass gave in (1869, Werke 2, p. 49) a counter example: for the non-negative

[4] Two simply connected surfaces can always be mapped one to the other, such that each point on the former moves continuously with the point on the latter...

[5] Weierstrass had taken Riemann's PhD thesis as vacation reading, and complained that for a function theorist like him, the methods of Riemann were hard to understand. Helmholtz then also borrowed the thesis, and said on their next meeting, that for him, Riemann's thoughts seemed to be completely natural and self-evident.

[6] To this end, one can often invoke a principle for finding a function that solves Laplace's equation, which Dirichlet has been using in his lectures over the past few years.

functional

$$\int_{-1}^{1} (x \cdot y')^2 \, dx \to \min \quad y(-1) = a,\ y(1) = b,$$

the function $y(x)$ must have a small derivative when x is large, to make the functional small. Hence the derivative can only be large when x is close to zero, and the minimum is achieved for the step function, which is not differentiable at $x = 0$. Weierstrass concludes

> "Die Dirichlet'sche Schlussweise führt also in dem betrachteten Falle offenbar zu einem falschen Resultat."[7]

But Riemann only answered "... meine Existenztheoreme sind trotzdem richtig"[8] and Helmholtz commented "Für uns Physiker bleibt das Dirichletsche Prinzip ein Beweis"[9].

4 The Schwarz Alternating Method

The entire mathematical world stood now in front of a big challenge, namely to show rigorously that for an arbitrary domain Ω, Laplace's equation $\Delta u = 0$ with prescribed boundary conditions $u = g$ on $\partial\Omega$ has a unique solution. For special domains, the answer had been known for quite some time: Poisson (1815) had found the solution formula for circular domains, and Fourier (1807) for rectangular domains using Fourier series. But the existence of solutions of Laplace's equation on arbitrary domains appeared hopeless !

It is at this moment, where Schwarz invented the first ever domain decomposition method [18]. His paper starts with the paragraph

> Die unter dem Namen Dirichlet'sches Princip bekannte Schlussweise, welche in gewissem Sinne als das Fundament des von Riemann entwickelten Zweiges der Theorie der analytischen Funktionen angesehen werden muss, unterliegt, wie jetzt wohl allgemein zugestanden wird, hinsichtlich der Strenge sehr begründeten Einwendungen, deren vollständige Entfernung, soviel ich weiss, den Anstrengungen der Mathematiker bisher nicht gelungen ist. [10]

Schwarz then invents the famous alternating Schwarz method to prove existence and uniqueness of the solution of Laplace's equation on a domain composed of a disk and a rectangle, as shown from the original publication in Figure 5 on the left. His alternating method is given by

---

[7] Dirichlet's reasoning apparently leads to an incorrect result in this case [8].

[8] ... my existence theorems nevertheless hold [8].

[9] For us physicists the Dirichlet principle remains a proof [8].

[10] The method of conclusion, which became known under the name Dirichlet Principle, and which in a certain sense has to be considered to be the foundation of the theory of analytic functions de-

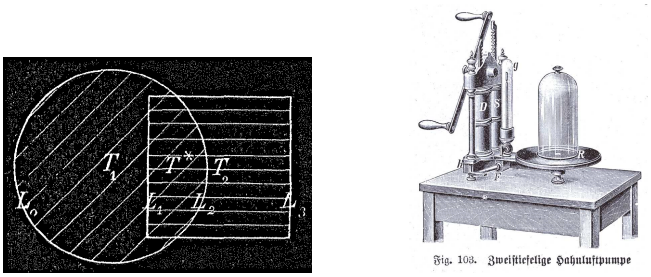


Fig. 5 Original drawing of Schwarz from 1870 on the left to explain his alternating method, and his physical interpretation of the method using a two level vacuum pump on the right

$$\begin{array}{rlrl} \Delta u_1^n = 0 & \text{in } T_1, & \Delta u_2^n = 0 & \text{in } T_2, \\ u_1^n = g & \text{on } L_0, & u_2^n = g & \text{on } L_3, \\ u_1^n = u_2^{n-1} & \text{on } L_2, & u_2^n = u_1^n & \text{on } L_1. \end{array} \tag{6}$$

Since the method only uses solutions of Laplace's equation on the disk and the rectangle, for which the proof of the Dirichlet principle did not pose any difficulties, the method is well defined. Schwarz then proved the convergence of his method to a limit that satisfies Laplaces equation as well in the composed domain. Adding other circles or rectangles Schwarz then proved recursively the Dirichlet principle for more and more complicated domains. This closed the gap in Riemann's proof.

Schwarz also gave an analogy of his alternating method with a physical device, as indicated on the right in Figure 5: a vacuum pump with two cylinders. In order to create a vacuum in the inner chamber, one has to alternatingly pump with the two cylinders, similar to the subdomain solves in the alternating method.

5 The Schwarz method as a computational tool

At the beginning of the 20th Century, Hilbert (see [6, 7]) finally managed, after a hard struggle, to establish a theory for *direct methods of variational calculus*, which later led to the Ritz-Galerkin method (see e.g. [5]). The Schwarz method thus lost completely its importance as a theoretical tool. Curiously, some other decades later, its importance for *practical computations* was discovered: in 1965, Miller states [14]:

> "Schwarz's method presents some intriguing possibilities for numerical methods. Firstly, quite simple explicit solutions by classical methods are often known for simple regions such as rectangles or circles. Also, better numerical solutions, from the standpoint of the computational work involved, are often known for certain types of regions than for others. By Schwarz's method, we may be able to extend these classical results and these computational advantages to more complicated regions."

---

veloped by Riemann, is subject to, like it is generally admitted now, very well justified objections, whose complete removal has eluded all efforts of mathematicians to the best of my knowledge.

Fundamental early contributions to the theory were by Sobolev [19], who gave a variational convergence proof for the case of elasticity, Mikhlin [13], with a variational proof for convergence for general elliptic operators, and then the sequence of publications by Lions [10, 11, 12]. The complete breakthrough as a computational method came with the introduction of the two level additive Schwarz method [1].

References

1. Maksymilian Dryja and Olof B. Widlund. An additive variant of the Schwarz alternating method for the case of many subregions. Technical Report 339, also Ultracomputer Note 131, Department of Computer Science, Courant Institute, 1987.
2. Leonard Euler. Principia motus fluidorum. *Novi Commentarii academiae scientiarum Petropolitanae*, 6:271–311, 1756.
3. Joseph Fourier. *Théorie analytique de la chaleur*. Firmin Didot, père et fils, 1822.
4. Martin J. Gander. Schwarz methods over the course of time. *ETNA*, 31:228–255, 2008.
5. Martin J. Gander and Gerhard Wanner. From Euler, Ritz, Galerkin to modern computing. *SIAM Rev.*, 54(4), 2012.
6. David Hilbert. Über das Dirichletsche Prinzip. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 8:184–188, 1900. Reprinted in 'Journal für die reine und angewandte Mathematik' Vol 129, 1905, pp. 63–67.
7. David Hilbert. Über das Dirichletsche Prinzip. *Mathematische Annalen*, 59:161–186, 1904.
8. Felix Klein. *Vorlesungen über die Entwicklung der Mathematik im 19. Jahrhundert*. Berlin, 1926. Reprinted New York 1950 and 1967.
9. Pierre Simon Laplace. *Traité de Mécanique Céleste*. De l'Imprimerie de Crapelet, Paris (an VII), 1799.
10. Pierre-Louis Lions. On the Schwarz alternating method. I. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42, Philadelphia, PA, 1988. SIAM.
11. Pierre-Louis Lions. On the Schwarz alternating method II: Stochastic interpretation and orders properties. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods*, pages 47–70, Philadelphia, PA, 1989. SIAM.
12. Pierre-Louis Lions. On the Schwarz alternating method III: A variant for nonoverlapping subdomains. In Tony F. Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, held in Houston, Texas, March 20-22, 1989*, pages 202–223, Philadelphia, PA, 1990. SIAM.
13. S.G. Mikhlin. On the Schwarz algorithm. *Doklady Akademii Nauk SSSR*, 77:569–571, 1951.
14. Keith Miller. Numerical analogs to the Schwarz alternating procedure. *Numer. Math.*, 7:91–103, 1965.
15. Isaac Newton. *Philosophiae Naturalis Principia Mathematica*. Juffu Societatis Regiae ac Typis Josephi Streater, Londini, 1687.
16. R. Remmert. *Funktionentheorie*. Springer, 1991.
17. Berhard Riemann. *Grundlagen für eine allgemeine Theorie der Functionen einer veränderlichen complexen Grösse*. PhD thesis, Göttingen, 1851. Werke p. 3–34, transcribed by D. R. Wilkins, April 2000.
18. H. A. Schwarz. Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, Mai 1870.
19. S. L. Sobolev. L'Algorithme de Schwarz dans la Théorie de l'Elasticité. *Comptes Rendus (Doklady) de l'Académie des Sciences de l'URSS*, IV((XIII) 6):243–246, 1936.
20. Gerhard Wanner. Kepler, Newton and numerical analysis. *Acta Numerica*, 19:561–598, 2010.

Solving large systems on HECToR using the 2-Lagrange multiplier methods

Anastasios Karangelis[1], Sébastien Loisel[1], and Chris Maynard[2]

1 Introduction

We consider the model problem,

$$-\Delta\tilde{u} = \tilde{f} \text{ in } \Omega \text{ and } \tilde{u} = 0 \text{ on } \partial\Omega. \tag{1}$$

In order to solve the problem numerically we discretize it by some suitable method[1] and as a result we get the system,

$$Au = f, \tag{2}$$

where A is a large symmetric and positive definite sparse matrix, f is the load vector and u is the desired discrete solution of our problem. Note that we use the notation $\tilde{u} = \tilde{u}(x)$ for the solution $\tilde{u} \in H_0^1(\Omega)$ and u for corresponding finite element coefficient vector.

We decompose our square model domain Ω into nonoverlapping rectangular subdomains $\Omega_1, \ldots, \Omega_p$ and we define the artificial interface $\Gamma = \Omega \cap (\bigcup_{i=1}^p \partial\Omega_i)$, such that $\Omega = \Gamma \cup \left(\bigcup_{k=1}^p \Omega_k\right)$ with disjoint unions. Although our numerical experiments are on a square, the analysis in [3, 5] applies to more general "shape-regular" domain decompositions and grids such as described in [8].

The local Robin subproblems are,

$$\begin{cases} -\Delta\tilde{u}_k = \tilde{f} & \text{in } \Omega_k, \\ \tilde{u}_k = 0 & \text{on } \partial\Omega_k \cap \partial\Omega, \\ (a + D_\nu)\tilde{u}_k = \tilde{\lambda}_k & \text{on } \partial\Omega_k \cap \Gamma; \end{cases} \tag{3}$$

where $a > 0$ is the Robin parameter, $k = 1, \ldots, p$ and D_ν denotes the directional derivative in the direction of the unit outwards normal vector ν of $\partial\Omega$, and $\tilde{\lambda}_k$ is the Robin data imposed on the "artificial interface" $\partial\Omega_k \cap \Gamma$.

We now discretize system (3) using the finite element method. This leads to linear systems of the form,

$$\begin{bmatrix} A_{Ik} & A_{I\Gamma k} \\ A_{\Gamma Ik} & A_{\Gamma\Gamma k} + aI \end{bmatrix} \begin{bmatrix} u_{Ik} \\ u_{\Gamma k} \end{bmatrix} = \begin{bmatrix} f_{Ik} \\ f_{\Gamma k} \end{bmatrix} + \begin{bmatrix} 0 \\ \lambda_k \end{bmatrix}. \tag{4}$$

[1] Dept. of Mathematics, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom, e-mail: {ak411}{S.Loisel}@hw.ac.uk .[2] EPCC, University of Edinburgh, Edinburgh EH9 3JZ, United Kingdom, e-mail: c.maynard@ed.ac.uk and Met Office, FitzRoy Road, Exeter, EX1 3PB, e-mail: christopher.maynard@metoffice.gov.uk

[1] In general this could be by finite elements or finite differences.

Here, the subscript I denotes nodes that are Interior to Ω_k, while the subscript Γ denotes nodes on $\Gamma \cap \partial\Omega_k$; this notation is consistent with existing literature, see [5], [8]. Using a Schur complement, we eliminate the interior nodes of equation (4) to get the equivalent system,

$$(S+aI)u_G = g+\lambda, \tag{5}$$

where $S = \mathrm{diag}\{S_1,\ldots,S_p\}$ with symmetric and semidefinite Schur complements $S_k = A_{\Gamma\Gamma k} - A_{\Gamma Ik}A_{IIk}^{-1}A_{I\Gamma k}$; the column vector $u_G = [u_{\Gamma 1}^T,\ldots,u_{\Gamma p}^T]^T$ is the multi-valued trace (with one value per interface vertex per adjacent subdomain), the Robin data are $\lambda = [\lambda_1^T,\ldots,\lambda_p^T]^T$ and the "accumulated fluxes" are $g_k = f_{\Gamma k} - A_{\Gamma Ik}A_{IIk}^{-1}f_{Ik}$.

We define the scaled "Robin-to-Dirichlet" map $Q = \mathrm{diag}\{Q_1,\ldots,Q_p\}$, where $Q_k = a(S_k+aI_k)^{-1}$ and (5) can be rewritten as,

$$au_G = Q(g+\lambda). \tag{6}$$

The multi-valued trace u_G can be interpreted as the multi-valued trace of a finite element function $\tilde{u}(x)$ which has jump discontinuities along Γ. For each vertex $x_j \in \Gamma$ on the interface, we define m_j to be the number of subdomains adjacent to x_j. A vertex with $m_j = 2$ is called a regular interface point while a vertex with $m_j > 2$ is called a cross point. The solution of (1) is continuous and so we must impose continuity on $\tilde{u}(x)$ (or equivalently, on its finite element trace vector u_G). To that end, we define K to be the orthogonal projection matrix which averages the function values for each interface vertex x_j; note that the range of K is precisely the space of continuous many-sided traces. Hence, u_G is continuous if and only if,

$$Ku_G = u_G. \tag{7}$$

Additionally we require the "fluxes" to match which is equivalent to

$$K(Su_G) = Kg. \tag{8}$$

1.1 Obtaining the S2LM and 2LM systems

From (5) and (7) we get that,

$$KQ(\lambda+g) = Q(\lambda+g), \tag{9}$$

and from (8) we get,

$$K(g+\lambda-Q(g+\lambda)) = Kg. \tag{10}$$

We add (9) and (10) to get the **symmetric 2-Lagrange multiplier system** (S2LM) ,

$$(Q-K)\lambda = -Qg. \tag{11}$$

Multiplying both sides of (11) on the left by $(I-2K)$, we get the corresponding **nonsymmetric 2-Lagrange system** (2LM),

$$(I-2K)(Q-K)\lambda = -(I-2K)Qg. \tag{12}$$

We now briefly summarize some known results about the 2-Lagrange methods and refer to [4], [5], [3] for details.

Theorem 1. *We define E to be the orthogonal projection onto the kernel of S. Assume that $\|EK\| < 1$. Then (11) is equivalent to (2).*

Theorem 2. *[4] The nonsymmetric 2-Lagrange system, $(I-2K)(Q-K)$ is an Optimised Schwarz Method (at least for two subdomains)*

The 2-Lagrange multiplier methods also have a coarse grid preconditioner,

$$P = I - EKE, \tag{13}$$

leading to the 2-level methods,

$$P^{-1}(Q-K)\lambda = -P^{-1}Qg, \tag{14}$$
$$P^{-1}(I-2K)(Q-K)\lambda = -P^{-1}(I-2K)Qg. \tag{15}$$

Theorem 3. *The optimized Robin parameter $a = \sqrt{s_{min}s_{max}}$, where s_{min} and s_{max} are the extremal eigenvalues of S. Moreover,*
The condition number for the 1-level methods is $O(h^{-1/2}H^{-3/2})$
The condition number for the 2-level methods is $O(H/h)^{1/2}$

2 Implementation of symmetric and nonsymmetric 2-Lagrange multiplier and large scale experiments on HECToR

The numerical experiments were run on HECToR, a Cray XE6 with 2816 compute nodes each comprising of two 16-core AMD Opeteron Interlagos processors. Each of the 16-core socket is coupled with a Cray Gemini routing and communications chip.

2.1 Implementation

We have implemented the symmetric and nonsymmetric 2LM methods in C using the PETSc library [1]. We implemented three matrices K, Q and the coarse grid preconditioner P. The matrices P,Q are implemented as PETSc shell matrices while the K matrix is assembled into a `seqaij` matrix. In other words, the matrix K is assembled into PETSc's parallel compressed row storage sparse matrix format, while

the matrices P and Q are not assembled but instead a matrix-vector multiplication routine is provided to PETSc. The matrices P and Q are not assembled because they are not sparse.

We use a PETSc parallel Krylov space solver on (14) or (15) as an "outer iteration". Each step of the outer iteration requires multiplying a given vector by the matrices P, Q, K. The matrix-vector product $K\lambda$ is a straightforward sparse matrix-dense vector product. The matrix-vector product $Q\lambda$ requires solving subdomain problems as per (4). These subdomain problems can in principle become large. Thus, (4) is solved using a PETSc sequential Krylov space solver (ie. a single-processor solver) on (4); this is an "inner iteration" which occurs at each step of the outer iteration. Hence the overall algorithm has an inner-outer iteration structure. In our test implementation, we use a finite difference implementation with a square domain and rectangular subdomains, with one domain assigned per MPI task with affinity to a single core.

2.1.1 The matrix K

The solution λ to the linear systems, (11) or (12) is a multi-valued trace, with one function value per artificial interface point per subdomain. In PETSc, the rows of λ are distributed such that the indices of the same domain are assigned to a single processor,

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_p \end{bmatrix}.$$

Each entry in λ corresponds to an artificial interface grid point. When two or more subdomains are adjacent, then some entries of λ correspond to the same artificial interface point.

Each processor lists the physical grid points on its artificial interface; this information is shared with neighboring subdomains using MPI explicitly. When solving subdomain problems, we work with small-dimensional local vectors. The Robin data λ_j on subdomain Ω_j has length $n_{\Gamma j}$; we write $\lambda_j = (\lambda_i^{(j)})_{i=1}^{n_{\Gamma j}}$. Mapping from the "local index" i to a "global offset" is achieved with the function $F_j(i) = i + \sum_{k<j} n_{\Gamma k}$. The size of the matrix K is $\sum_{k=1}^{p} n_{\Gamma k}$. Given this information, each processor is able to assemble its own rows of K.

2.1.2 The matrix Q

We begin by showing that the matrix-vector product $\lambda_k \mapsto Q_k\lambda_k$ can be computed by solving a local sparse problem. Setting $f = 0$ (and hence $g = 0$) in (4) and (5) shows that $Q_k\lambda_k = au_{\Gamma k}$, where $u_{\Gamma k}$ is defined by,

$$\begin{bmatrix} A_{IIk} & A_{I\Gamma k} \\ A_{\Gamma Ik} & A_{\Gamma\Gamma k} + aI \end{bmatrix} \begin{bmatrix} u_{Ik} \\ u_{\Gamma k} \end{bmatrix} = \begin{bmatrix} 0 \\ \lambda_k \end{bmatrix}. \tag{16}$$

Thus, in order to calculate the matrix-vector product $Q\lambda$, each processor solves the Robin local problem (16) and outputs $Q_k\lambda_k = au_{\Gamma k}$.

The local problem (16) can in principle be solved using eg. a Cholesky decomposition. However, we found that using a Cholesky decomposition leads to large amounts of fill-in and poor performance. Thus, we solve the local problem (16) using the Conjugate Gradient method with relative convergence tolerance 1e-10 and absolute convergence tolerance 1e-9. For the local problem (16), we use the incomplete Cholesky ICC(ℓ) preconditioner [2]. The incomplete Cholesky preconditioner is a compromise between higher fill-in (leading in the limit to a direct solver) and lower fill-in (leading in the limit to a diagonal preconditioner). We found that a "factor level" $\ell = 10$ gives better overall performance for our problem sizes.

2.1.3 The preconditioner P

The coarse grid preconditioner matrix P defined in (13) is in principle an enormous parallel matrix. Nevertheless, we will describe an efficient way to compute the matrix-vector product $\lambda \mapsto P^{-1}\lambda$ efficiently on a single processor (with some global communication). For $j = 1, \ldots, p$ we denote $n_{\Gamma j}$ the number of vertices on the artificial interface $\partial\Omega_j \cap \Gamma$ and we define the matrix $J := \mathrm{diag}(\frac{1}{\sqrt{n_{\Gamma 1}}}\mathbf{1}_{n_{\Gamma 1}}, \ldots, \frac{1}{\sqrt{n_{\Gamma p}}}\mathbf{1}_{n_{\Gamma p}})$ where $\mathbf{1}_j$ denotes the jth dimensional column vector of ones. The columns of J span the "coarse space" of piecewise constant functions, which are constant on each local artificial interface $\Gamma_k = \partial\Omega_k \cap \Gamma$. The coarse space for the preconditioner (13) is the kernel of S, which is contained in the column span of J. Thus, we define $E := JJ^T$ and,

$$P^{-1} := (I - EKE)^{-1} = I - JJ^T - J(\overbrace{J^TKJ - I}^{L})^{-1}J^T.$$

Note that although P^{-1} is dense, we can compute $\lambda \mapsto P^{-1}\lambda$ efficiently, in a matrix-free way, via the formula $P^{-1}\lambda = \lambda - J(J^T\lambda) - J(L^{-1}(J^T\lambda))$.

Given the assembled parallel sparse matrix J and its transpose J^T and the assembled (sparse) local matrix L, the algorithm for computing the matrix-vector product $\lambda \mapsto P^{-1}\lambda$ in a matrix-free way is as follows:

(i) Given λ, compute the p-dimensional "coarse" vector $\lambda_c = J^T\lambda$ and collect its entries on a single processor as a sequential vector.
(ii) Define u_c by solving the local, sparse linear problem $Lu_c = \lambda_c$.
(iii) Output $P^{-1}\lambda = \lambda - J\lambda_c - Ju_c$. Note that multiplication by J involves broadcasting the small local vectors λ_c and u_c to large parallel vectors $J\lambda_c$ and Ju_c.

Table 1: Iteration counts for S2LM.

# Procs.	Domain size 100^2	300^2	1000^2	3000^2
64	216	409	952	2472
256	173	316	782	1753
1024	144	220	411	1090
4096	-	-	301	665

Table 2: Iteration counts for 2LM.

# Procs.	Domain size 100^2	300^2	1000^2	3000^2	10000^2
64	30	58	114	229	-
256	37	35	72	135	-
1024	47	44	42	76	-
4096	-	-	53	50	82

2.1.4 The outer solve

The implementations of the shell matrices P and Q and the assembly of the sparse matrix K have been described. Building on these base implementations, we further form the shell matrices $\lambda \mapsto (Q-K)\lambda$ (implemented as `QminKmul`) and $\lambda \mapsto (I-2K)(Q-K)\lambda$ (implemented as `Imin2KQminKmul`). The PETSc library enables us to use a variety of different solvers. For the outer iteration we experimented with the Generalized Minimal Residual `KSPGMRES` and the Flexible Generalised Minimal Residual method `KSPFGMRES` on shell matrices `QminKmul` and `Imin2KmulQminK`, with the preconditioner P. For the `KSPFGMRES` solver we set the relative convergence tolerance $1e-7$ and the absolute convergence tolerance $1e-6$.

Recall that GMRES is an iterative method that computes the approximate solution $x_k \in x_0 + \text{span}\{r_0, Ar_0, \ldots A^k r_0\}$ which minimizes the residual norm $\|b-Ax_k\|_2$. The efficient implementation of the least-squares problem relies on the identity

$$AV_k = V_{k+1}\tilde{H}_k, \tag{17}$$

where V_k is an orthonormal basis of the Krylov space and $\tilde{H}_k$. is an upper Hessenberg matrix; cf. [7] for details. The Flexible GMRES algorithm [6] replaces (17) by,

$$AZ_m = V_{k+1}\tilde{H}_k, \tag{18}$$

and allows one to vary the preconditioner at each iteration, which required testing since our matrix-vector products are inexact.

2.1.5 Experiments at large scale

Results for the iteration counts of the S2LM and 2LM methods are presented. In both cases the Flexible GMRES algorithm for the outer solver and the Conjugate Gradient algorithm for the inner solver were used. The preconditioner for the outer solve is the shell matrix P, while the preconditioner for the inner solve is the incomplete Cholesky ICC(10) of (16). The other parameters for the solvers have been specified in sections 2.1.2 and 2.1.4.

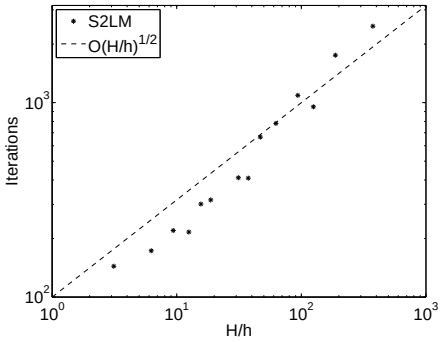


Fig. 1: Scaling of S2LM.

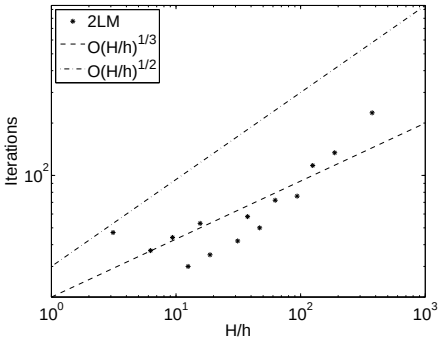


Fig. 2: Scaling of 2LM.

The implementation used here is limited to a square domain in two dimensions using a finite difference discretization. This choice was made entirely for the simplicity of implementation. The domains vary from 100^2 to 10000^2 grid points (and hence the largest problem has 10^8 degrees of freedom). These domains are partitioned into 64 to 4096 subdomains, which again is limited to a square number. This domain decomposition is mapped to the MPI decomposition on HECToR.

The symmetric (11) and nonsymmetric systems (12) are solved, with tolerances as in section 2.1.4; the outer iteration counts are reported in Tables 1 and 2. The computational cost per outer iteration for a fixed domain and subdomain is constant. The inner iterations are not reported as the ICC preconditioner is used for simplicity rather than the optimal multigrid which would be used as first choice in a production implementation. In addition to these raw iteration counts, we also plot the scaling of the methods against the ratio H/h in Figs. 1 and 2.

The S2LM performance is well explained by the condition number estimate of Theorem 3. Indeed, the S2LM matrix is symmetric and indefinite and for such systems, one can show that the number of iterations is bounded by a quantity proportional to the condition number. This bound is only sharp when the spectrum of the matrix is perfectly symmetric about the origin. We find that some of our smaller systems perform slightly better than this theoretical estimate.

The 2LM performance appears to be between $O(H/h)^{1/3}$ and $O(H/h)^{1/2}$. The 2LM matrix is nonsymmetric. For nonsymmetric matrices, the condition number does not necessarily predict the performance of the GMRES algorithm. However, in our case, we find that the condition number explains well the performance of the algorithm and that we further get "Krylov acceleration" – the performance may be almost as good as $O(H/h)^{1/3}$.

3 Conclusions

We have provided a large-scale implementation of the 2-Lagrange multiplier methods with cross points and a coarse grid correction, which we have tested on the HECToR supercomputer. Our experiments confirm the good scaling properties of the 2-Lagrange multiplier methods. In the future, we intend to improve our implementation to further explore the scaling to the largest systems.

Acknowledgements We gratefully acknowledge the support of the Centre for Numerical Algorithms and Intelligent Software (EPSRC EP/G036136/1). This work made use of the facilities of HECToR, the UKs national high-performance computing service, which is provided by UoE HPCx Ltd. at the University of Edinburgh, Cray Inc and NAG Ltd., and funded by the Office of Science and Technology through EPSRCs High End Computing Programme.

References

1. Balay, S., Brown, J., Buschelman, K., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc Web page (2012). Http://www.mcs.anl.gov/petsc
2. Chan, T.F., Van Der Vorst, H.A.: Approximate and incomplete factorizations. Parallel Numerical Algorithms, ICASE/LaRC Interdisciplinary Series in Science and Engineering pp. 167–202 (1997)
3. Drury, S.W., Loisel, S.: Sharp condition number estimates for the symmetric 2-lagrange multiplier method. Domain Decomposition Methods in Science and Engineering XX Lecture Notes in Computational Science and Engineering **91**, 255–261 (2013)
4. Farhat, C., Roux, F.X.: A method of finite element tearing and interconnecting and its parallel solution algorithm. Internat. J. Numer. Methods Engrg **32**, pp. 1205–1227 (1991)
5. Loisel, S.: Condition number estimates for the nonoverlapping optimized schwarz method and the 2-lagrange multiplier method for general domains and cross points. SIAM Journal on Numerical Analysis **51**:6, 30623083 (2013)
6. Saad, Y.: A flexible inner-outer preconditioned gmres algorithm. SIAM Journal on Scientific Computing **14**, 461–469 (1993)
7. Saad, Y., Schultz, M.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on scientific and statistical computing **7**(3), 856–869 (1986)
8. Toselli, A., Widlund, O.B.: Domain Decomposition Methods – Algorithms and Theory, vol. volume 34 of Springer Series in Computational Mathematics. Springer Berlin Heidelberg (2005)

Coupled Finite and Boundary Element Methods for Vibro–Acoustic Interface Problems

Arno Kimeswenger[1], Olaf Steinbach[1], and Gerhard Unger[1]

1 Vibro–Acoustic Interface Problem

As a vibro–acoustic interface model problem we consider a three–dimensional elastic body, e.g., a submarine, which is completely immersed in a full space acoustic region, e.g., water [5]. Other applications that we have in mind are the sound radiation of passenger car bodies, where the acoustic region is bounded, or of partially immersed bodies such as ships, where the acoustic region is a half space [2].

In this paper, we consider both a direct simulation of the interface problem by using a symmetric coupled finite and boundary element approach, and an eigenvalue analysis to determine the eigenmodes of the coupled system. The time–harmonic vibrating structure in Ω_s is modeled by the Navier equations in the frequency domain, while the acoustic fluid in the unbounded exterior domain Ω_f is described by the Helmholtz equation,

$$-\rho_s\omega^2\mathbf{u}-\mu\Delta\mathbf{u}-(\lambda+\mu)\operatorname{grad}\operatorname{div}\mathbf{u}=\mathbf{f}\text{ in }\Omega_s,\quad \kappa^2 p+\Delta p=0\text{ in }\Omega_f. \tag{1}$$

In (1), λ and μ are the Lamé parameters, ρ_s and ρ_f are the densities of the structure and of the acoustic fluid, respectively, ω is the frequency, and $\kappa=\omega/c\in\mathbb{R}$ is the wave number. Note that $\Omega_s\subset\mathbb{R}^3$ is in general a bounded, multiple connected domain with an interior boundary $\Gamma_I=\overline{\Gamma}_D\cup\overline{\Gamma}_N$, $\Gamma_D\cap\Gamma_N=\emptyset$, see Fig. 1, where boundary conditions of Dirichlet and Neumann type are given,

$$\mathbf{u}=\mathbf{g}_D\text{ on }\Gamma_D,\quad T\mathbf{u}:=\lambda(\operatorname{div}\mathbf{u})\,\mathbf{n}+2\mu\frac{\partial}{\partial n}\mathbf{u}+\mu\,\mathbf{n}\times\operatorname{curl}\mathbf{u}=\mathbf{g}_N\text{ on }\Gamma_N. \tag{2}$$

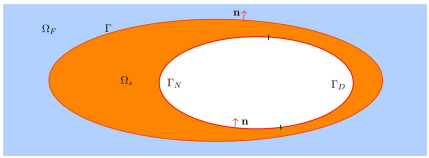


Fig. 1 Computational domain and boundary conditions

In addition to the partial differential equations (1) and the boundary conditions (2) we consider transmission conditions on $\Gamma=\overline{\Omega}_s\cap\overline{\Omega}_f$,

$$T\mathbf{u}+p\mathbf{n}=\mathbf{0},\quad \rho_f\omega^2\mathbf{u}\cdot\mathbf{n}=\mathbf{n}\cdot\nabla p\quad\text{on }\Gamma. \tag{3}$$

[1] Institut für Numerische Mathematik, TU Graz, Steyrergasse 30, 8010 Graz, Austria, e-mail: {arno.kimeswenger}{o.steinbach}{gerhard.unger}@tugraz.at

Finally, p has to satisfy a radiation condition at infinity,

$$\lim_{r\to\infty}\int_{|x|=r}\left|\frac{\partial}{\partial n_x}p(x)-i\kappa p(x)\right|^2 ds_x = 0. \tag{4}$$

For complex wave numbers $\kappa\in\mathbb{C}$ with $\mathrm{Im}(\kappa)<0$, instead of (4) one has to use a radiation condition in terms of spherical Hankel functions in order to describe outgoing waves, see [12].

The aim of this paper is to derive and to discuss a symmetric coupled finite and boundary element formulation which is stable for almost all frequencies $\omega\in\mathbb{R}$, and to characterize all eigenfrequencies $\omega\in\mathbb{C}$ which imply non–trivial solutions of the homogeneous transmission problem (1)–(4), i.e. for $\mathbf{f}=\mathbf{0}$, $\mathbf{g}_D=\mathbf{0}$, $\mathbf{g}_N=\mathbf{0}$. In fact, in this case only one of the three following situations may appear [9]:

i. A real eigenfrequency $\omega\in\mathbb{R}$ implies $p=0$, and any non–trivial solution $\mathbf{u}$ is a so–called Jones mode satisfying $T\mathbf{u}=\mathbf{0}$ and $\mathbf{u}\cdot\mathbf{n}=0$ on Γ [6].
ii. A complex value $\omega\in\mathbb{C}$ with $\mathrm{Im}(\omega)>0$ implies $\mathbf{u}=\mathbf{0}$ and $p=0$.
iii. If $\omega\in\mathbb{C}\backslash\mathbb{R}$ is an eigenfrequency, then $\mathrm{Im}(\omega)<0$.

In the low frequency regime one may consider an approximation of the Helmholtz equation in (1) by the Laplace equation, for related coupled finite and boundary element formulations, see [10].

2 Coupled finite and boundary element methods

The symmetric coupling [4] of finite and boundary elements for the transmission boundary value problem (1)–(4) relies on the standard variational formulation of the Navier equations in Ω_s, and the use of the exterior Calderon projection of boundary integral equations [13] to describe the solution of the Helmholtz equation in Ω_f. The resulting variational formulation is to find $\mathbf{u}\in[H^1(\Omega_s)]^3$, $\mathbf{u}=\mathbf{g}_D$ on Γ_D, such that

$$\begin{aligned}&\int_{\Omega_s}\Big[2\mu\, e(\mathbf{u}):e(\mathbf{v})+\lambda\,\mathrm{div}\mathbf{u}\,\mathrm{div}\mathbf{v}\Big]dx-\rho_s\omega^2\int_{\Omega_s}\mathbf{u}\cdot\mathbf{v}\,dx \\ &-\rho_f\omega^2\langle V_\kappa[\mathbf{u}\cdot\mathbf{n}],\mathbf{v}\cdot\mathbf{n}\rangle_\Gamma+\langle(\frac{1}{2}I+K_\kappa)p,\mathbf{v}\cdot\mathbf{n}\rangle_\Gamma=\int_{\Omega_s}\mathbf{f}\cdot\mathbf{v}\,dx+\int_{\Gamma_N}\mathbf{g}_N\cdot\mathbf{v}\,ds_x\end{aligned} \tag{5}$$

is satisfied for all $\mathbf{v}\in[H^1(\Omega_s)]^3$, $\mathbf{v}=\mathbf{0}$ on Γ_D, where $p\in H^{1/2}(\Gamma)$ is a solution of the hypersingular boundary integral equation

$$\frac{1}{\rho_f\omega^2}D_\kappa p+(\frac{1}{2}I+K'_\kappa)[\mathbf{u}\cdot\mathbf{n}]=0\quad\text{on }\Gamma. \tag{6}$$

The boundary integral operators are defined as, for $x\in\Gamma$,

$$(V_\kappa q)(x) = \int_\Gamma U^*_\kappa(x,y)q(y)ds_y, \qquad (K_\kappa p)(x) = \int_\Gamma \frac{\partial}{\partial n_y} U^*_\kappa(x,y)p(y)ds_y,$$
$$(K'_\kappa q)(x) = \int_\Gamma \frac{\partial}{\partial n_x} U^*_\kappa(x,y)q(y)ds_y, \quad (D_\kappa p)(x) = -\frac{\partial}{\partial n_x}\int_\Gamma \frac{\partial}{\partial n_y} U^*_\kappa(x,y)p(y)ds_y,$$

where the Helmholtz fundamental solution is

$$U^*_\kappa(x,y) = \frac{1}{4\pi}\frac{e^{i\kappa|x-y|}}{|x-y|} \quad \text{for } x,y \in \mathbb{R}^3.$$

For the mapping properties of all boundary integral operators, see, for example, [13]. In particular, the hypersingular integral operator $D_\kappa : H^{1/2}(\Gamma) \to H^{-1/2}(\Gamma)$ is coercive and injective, if κ^2 is not an eigenvalue of the related interior Neumann eigenvalue problem of the Laplace operator in $\mathbb{R}^3\backslash\overline{\Omega}_f$. However, since we are using a direct approach we find $(\frac{1}{2}I + K'_\kappa)[\mathbf{u}\cdot\mathbf{n}] \in \operatorname{Im} D_\kappa$ even in the case when κ^2 is an eigenvalue of the interior Neumann eigenvalue problem with a related eigensolution $p_{\kappa^2|\Gamma} \in H^{1/2}(\Gamma)$ [14], i.e.

$$-\Delta p_{\kappa^2} = \kappa^2 p_{\kappa^2} \text{ in } \mathbb{R}^3\backslash\overline{\Omega}_f, \quad \frac{\partial}{\partial n} p_{\kappa^2} = 0 \text{ on } \Gamma.$$

The general solution of the hypersingular boundary integral equation (6) is then given by

$$p = -\rho_f \omega^2 D_\kappa^{-1}(\frac{1}{2}I + K'_\kappa)[\mathbf{u}\cdot\mathbf{n}] + \alpha p_{\kappa^2}, \tag{7}$$

where D_κ^{-1} has to be understood as a pseudoinverse. Note that $\alpha \in \mathbb{R}$ is an arbitrary constant. However, when inserting the solution p as given in (7) into the variational formulation (5), we have to evaluate

$$\begin{aligned}(\frac{1}{2}I + K_\kappa)p &= -\rho_f\omega^2(\frac{1}{2}I + K_\kappa)D_\kappa^{-1}(\frac{1}{2}I + K'_\kappa)[\mathbf{u}\cdot\mathbf{n}] + \alpha(\frac{1}{2}I + K_\kappa)p_{\kappa^2} \\ &= -\rho_f\omega^2(\frac{1}{2}I + K_\kappa)D_\kappa^{-1}(\frac{1}{2}I + K'_\kappa)[\mathbf{u}\cdot\mathbf{n}]\end{aligned}$$

due to $\ker D_\kappa = \ker(\frac{1}{2}I + K_\kappa)$. In fact, the Poincaré–Steklov operator

$$T_\kappa := V_\kappa + (\frac{1}{2}I + K_\kappa)D_\kappa^{-1}(\frac{1}{2}I + K'_\kappa) : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma)$$

is well defined for all frequencies ω. Hence we conclude the variational problem to find $\mathbf{u} \in [H^1(\Omega_s)]^3$, $\mathbf{u} = \mathbf{g}_D$ on Γ_D, such that

$$\begin{aligned}&\int_{\Omega_s}\Big[2\mu\, e(\mathbf{u}) : e(\mathbf{v}) + \lambda \operatorname{div}\mathbf{u} \operatorname{div}\mathbf{v}\Big]dx \\ &\quad -\omega^2\left[\rho_s\int_{\Omega_s}\mathbf{u}\cdot\mathbf{v}\,dx + \rho_f\langle T_\kappa[\mathbf{u}\cdot\mathbf{n}], \mathbf{v}\cdot\mathbf{n}\rangle_\Gamma\right] = \int_{\Omega_s}\mathbf{f}\cdot\mathbf{v}\,dx + \int_{\Gamma_N}\mathbf{g}_N\cdot\mathbf{v}\,ds_x\end{aligned} \tag{8}$$

is satisfied for all $\mathbf{v} \in [H^1(\Omega_s)]^3$, $\mathbf{v} = \mathbf{0}$ on Γ_D. Since the bilinear form which is related to the variational formulation (8) is coercive, injectivity ensures unique solvability of the variational problem (8), see also [8, 9].

Theorem 1. *Assume that $\omega \in \mathbb{R}$ is not a Jones frequency. Then there exists a unique solution* $\mathbf{u}$ *of the variational problem (8).*

Remark 1. Although boundary value problems of the exterior Helmholtz equation are unique solvable, related boundary integral equations may suffer from spurious modes which correspond to solutions of related interior eigenvalue problems for the Laplacian. Formulations which are stable for all frequencies, are usually based on complex linear combinations of different boundary integral operators, see, e.g., [2, 9]. However, when using a direct boundary integral approach as presented here, this also leads to a stable formulation, see [14] for a further discussion.

In what follows we consider a frequency $\omega \in \mathbb{R}$ which is not a Jones mode. If the displacement field $\mathbf{u}$ is known as the unique solution of the variational problem (8), we may use the boundary integral equation (6) to determine the pressure p. In the case when κ^2 is an eigenvalue of the interior Neumann eigenvalue problem, the solution p as given in (7) is not unique. However, using the transmission conditions (3) we find

$$p = -T\mathbf{u} \cdot \mathbf{n}, \tag{9}$$

in fact $(\mathbf{u}, p)$ is the unique solution of the coupled variational formulation (5). The representation (9) can be used to modify the boundary integral equation (6) to obtain a formulation which admits a unique solution p for all frequencies, for example we may consider the boundary integral equation

$$\Big[\frac{1}{\rho_f \omega^2} D_\kappa + i\eta \widetilde{D}_0\Big] p + (\frac{1}{2} I + K'_\kappa)[\mathbf{u} \cdot \mathbf{n}] + i\eta \widetilde{D}_0 (T\mathbf{u} \cdot \mathbf{n}) = 0 \quad \text{on } \Gamma,$$

where $\widetilde{D}_0$ is the stabilized hypersingular boundary integral operator of the Laplacian [13], and $\eta \in \mathbb{R}$ is some parameter to be chosen. For simplicity of the presentation we only consider the discretization of the variational formulation (8) by using piecewise linear finite elements which are defined with respect to some admissible triangulation of Ω_s, and by using piecewise linear boundary elements on Γ. This leads to the linear system

$$\begin{pmatrix} K_h^{\mathrm{FEM}} - \omega^2[\rho_s M_h^{\mathrm{FEM}} + \rho_f N_h^\top V_h^{\mathrm{BEM}} N_h] & N_h^\top (\frac{1}{2} M_h^{\mathrm{BEM}} + K_h^{\mathrm{BEM}}) \\ (\frac{1}{2} M_h^{\mathrm{BEM},\top} + K_h'^{\,\mathrm{BEM}}) N_h & \frac{1}{\omega^2 \rho_f} D_h^{\mathrm{BEM}} \end{pmatrix} \begin{pmatrix} \underline{u} \\ \underline{p} \end{pmatrix} = \begin{pmatrix} \underline{f} \\ \underline{0} \end{pmatrix}.$$

Here, K_h^{FEM} and M_h^{FEM} are the finite element stiffness and mass matrices, respectively, and V_h^{BEM}, M_h^{BEM}, K_h^{BEM}, and D_h^{BEM} are the Galerkin boundary element matrices, see, e.g., [11], and N_h corresponds to the application of the normal component, $\mathbf{u} \cdot \mathbf{n}$. From the standard theory, e.g., [13], we expect a second order of convergence when measuring the error $\|\mathbf{u} - \mathbf{u}_h\|_{L_2(\Omega_s)}$. Although the pressure p on the boundary Γ may

not be unique, the computation of the pressure p in Ω_f by means of the exterior representation formula

$$\widetilde{p}(x) = -\rho_f \omega^2 \int_\Gamma U^*_\kappa(x,y)[\mathbf{u}_h(y)\cdot\mathbf{n}_y]\,ds_y + \int_\Gamma \frac{\partial}{\partial n_y} U^*_\kappa(x,y)\,p_h(y)\,ds_y \quad \text{for } x \in \Omega_f$$

is unique, and we conclude a second order convergence of the pointwise error [13].

As a numerical example for the direct simulation we consider the Neumann boundary value problem (1)–(4) with

$$\Omega_s := \left\{x \in \mathbb{R}^3 : 0.8 < |x| < 1\right\}, \quad \Omega_f := \left\{x \in \mathbb{R}^3 : 1 < |x|\right\},$$

where the exact solution is given by, $r = |x|$,

$$p(x) = \frac{e^{i\kappa r}}{r} \quad \text{for } r > 1, \quad u(r) = [c_1 u_1(r) + c_2 u_2(r)]e_r \quad \text{for } r \in (0.8, 1),$$

and

$$u_1(r) = -\frac{\sqrt{\lambda+2\mu}\cos\frac{r\sqrt{\rho_s}\omega}{\sqrt{\lambda+2\mu}}}{r\sqrt{\rho_s}\omega} + \frac{(\lambda+2\mu)\sin\frac{r\sqrt{\rho_s}\omega}{\sqrt{\lambda+2\mu}}}{r^2\rho_s\omega^2},$$
$$u_2(r) = -\frac{\sqrt{\lambda+2\mu}\sin\frac{r\sqrt{\rho_s}\omega}{\sqrt{\lambda+2\mu}}}{r\sqrt{\rho_s}\omega} - \frac{(\lambda+2\mu)\cos\frac{r\sqrt{\rho_s}\omega}{\sqrt{\lambda+2\mu}}}{r^2\rho_s\omega^2}.$$

Note that the constants c_1 and c_2 have to be chosen accordingly to satisfy the transmission conditions (3). The material constants are given as $E = 105 \cdot 10^9\, N/m^2$, $\nu = 0.34$, while the densities of the structure and of the fluid are chosen as $\rho_s = 1000\, kg/m^3$ and $\rho_f = 4500\, kg/m^3$, respectively. Recall that the speed of sound is $c = 1484 m/s$. As frequency we have chosen $\omega = 3090\, s^{-1}$ which corresponds to an eigenfrequency of the hypersingular boundary integral operator D_κ. In Table 1 we present the relative errors of the displacement field both in the $L_2(\Omega)$ and in the energy norm, where we observe quadratic and linear convergence, as predicted. In addition, we also give the pointwise error for the pressure which is evaluated in $\widehat{x} = (2,0,0)^\top$, again we observe a quadratic convergence as predicted [13].

Table 1 Convergence of the FEM/BEM approach for direct simulation

N_{FEM}	$\frac{\|u-u_h\|_{L_2(\Omega_s)}}{\|u\|_{L_2(\Omega)}}$	$\frac{\|u-u_h\|_{H^1(\Omega_s)}}{\|u\|_{H^1(\Omega)}}$	$\lvert p(\widehat{x}) - \widetilde{p}(\widehat{x})\rvert$
1948	9.93 –2	2.56 –1	5.37 –2
15584	2.71 –2	1.45 –1	1.44 –2
124672	7.27 –3	7.62 –2	3.69 –3

3 Eigenvalue analysis

In this section we discuss the solution of the eigenvalue problem which is related to the transmission problem (1)–(4). Based on the coupled formulation (8) of the transmission problem the following related eigenvalue problem is considered: Find $(\omega, \mathbf{u}, p)$ with $(\mathbf{u}, p) \neq (\mathbf{0}, 0)$ such that

$$A(\omega)\begin{pmatrix}\mathbf{u}\\ p\end{pmatrix} := \begin{pmatrix} -\omega^2\rho_S M_S + K_S - \rho_f\omega^2 N^* V_\kappa N & N^*(\frac{1}{2}I + K_\kappa) \\ (\frac{1}{2}I + K'_\kappa)N & \frac{1}{\omega^2\rho_f} D_\kappa \end{pmatrix}\begin{pmatrix}\mathbf{u}\\ p\end{pmatrix} = \begin{pmatrix}\mathbf{0}\\ 0\end{pmatrix}, \tag{10}$$

where M_S represents the mass term and K_S the stiffness term of the structure, and $N\mathbf{u} = \mathbf{u}_{|\Gamma} \cdot \mathbf{n}$. The boundary integral operators depend nonlinearly on the wave number $\kappa = \omega/c$, hence (10) is a nonlinear eigenvalue problem in ω. For the eigenvalue problem (10), in addition to the requested eigenvalues we also obtain eigenvalues which correspond to the Laplacian with a Neumann boundary condition. However, in practice the latter can be filtered out very easily.

A Galerkin finite and boundary element discretization of (10) results in a nonlinear matrix eigenvalue problem of the form

$$A_h(\omega_h)\begin{pmatrix}\underline{u}\\ \underline{p}\end{pmatrix} = \begin{pmatrix}\underline{0}\\ \underline{0}\end{pmatrix}. \tag{11}$$

A rigorous numerical analysis of the Galerkin eigenvalue problem (11) can be carried out within the framework of the concept of eigenvalue problems for holomorphic Fredholm operator-valued functions [15] and will be addressed in a forthcoming paper. This concept provides comprehensive convergence results which include error estimates for the eigenvalues and eigenspaces.

For the numerical solution of (11) we use the contour integral method [1]. This method is suitable for the extraction of all eigenvalues which lie inside of a predefined contour in the complex plane. An alternative approach for the numerical solution of the nonlinear eigenvalue problem (11) which is based on polynomial interpolation is presented in [3].

As a numerical example we consider the Neumann eigenvalue problem for the spherical shell $\Omega_S := \{x \in \mathbb{R}^3 : 4.95 < |x| < 5\}$ and for the fluid domain $\Omega_f := \{x \in \mathbb{R}^3 : |x| > 5\}$. For this example analytical approximations of the eigenvalues are derived in [7]. The material constants for the shell are $E = 207 \cdot 10^9\, N/m^2$, $\nu = 0.3$ and $\rho_S = 7669\, kg/m^3$. For the surrounding fluid, we choose $c = 1483.24\, m/s$. As ansatz spaces for the Galerkin eigenvalue problem (11) we use piecewise linear finite elements and piecewise linear boundary elements as in the previous section. The eigenvalues of practical interest are those which are lying close to the real axis, since the imaginary part of an eigenvalue corresponds to the damping of the related eigenfunction in time. As domain of interest for the eigenfrequencies $f = \omega/(2\pi)$ we have chosen the strip $\{f \in \mathbb{C} : 1 < \mathrm{Re}(f) < 90,\ -5 < \mathrm{Im}(f) < 5\}$. In this domain two analytical approximations are given in [7]. The results of the contour integral method are presented in Table 2 for different meshes. The approximations of the

eigenvalues on the two finest mesh levels match well with the analytical approximations.

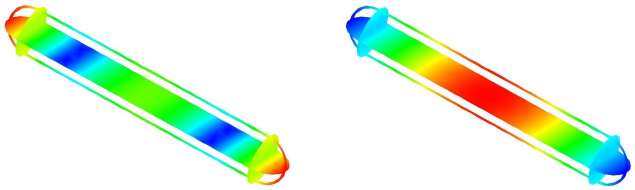
Table 2 Approximations of the two smallest non–zero eigenvalues $f = \omega/(2\pi)$

h/dof	0.5/8794	0.25/36792	0.15/109455	anal. approx.
	(58.19,-1.44)	(55.82,-1.18)	(55.65,-1.16)	56.02
	(58.26,-1.45)	(55.84,-1.18)	(55.66,-1.16)	
	(58.50,-1.48)	(55.84,-1.18)	(55.66,-1.16)	
	(58.62,-1.50)	(56.03,-1.20)	(55.78,-1.18)	
	(58.96,-1.54)	(56.04,-1.21)	(55.78,-1.18)	
	(83.61,-1.00)	(71.47,-0.32)	(70.45,-0.31)	70.52
	(83.73,-1.03)	(71.53,-0.32)	(70.53,-0.31)	
	(84.51,-1.08)	(71.63,-0.32)	(70.53,-0.31)	
	(85.10,-1.14)	(71.63,-0.32)	(70.54,-0.31)	
	(85.47,-1.16)	(71.72,-0.33)	(70.60,-0.31)	
	(85.94,-1.18)	(71.74,-0.33)	(70.61,-0.31)	
	(87.96,-1.37)	(71.80,-0.34)	(70.62,-0.32)	

4 Conclusions

The symmetric formulation of finite and boundary element methods for vibro–acoustic interface problems turns out to be stable for almost all freqencies. If we exclude Jones frequencies, no spurious modes appear. In fact, we can avoid the use of combined boundary integral equation formulations such as Brakhage/Werner and Burton/Miller, see, e.g., [2, 14], which require sufficient smoothness of the coupling interface. For the acceleration of the numerical simulations one may use fast boundary element methods such as the adaptive cross approximation [11] or the fast multipole method [2]. In addition, the design of appropriate preconditioned iterative solvers is a challenging task not only for the direct simulation. In fact, the contour integral method allows an reliable and accurate computation of eigenvalues within a given domain of interest, without any knowledge on the number and on the position of eigenvalues. Applications of the proposed methodologies include the simulation and eigenvalue analysis of ships, see Fig. 2 for a simplified model of a submarine made of titanium. The length is $12m$, its diameter $2m$, and its wall thickness $0.1m$. The first eigenfrequency is $f = 52.12 - 0.007i$, the related eigensolution is given in Fig. 2. This simulation was done by using 67.145 tetrahedral finite elements and 17.372 triangular boundary elements, which results in 74.523 global degrees of freedom.

Fig. 2 Real and imaginary part of an eigensolution of a simplified submarine



References

1. Beyn, W.J.: An integral method for solving nonlinear eigenvalue problems. Linear Algebra Appl. **436**, 3839–3863 (2012)
2. Brunner, D., Of, G., Junge, M., Steinbach, O., Gaul, L.: A fast BE–FE coupling scheme for partly immersed bodies. Int. J. Numer. Meth. Engrg. **81**, 28–47 (2010)
3. Effenberger, C., Kressner, D., Steinbach, O., Unger, G.: Interpolation–based solution of a nonlinear eigenvalue problem in fluid–structure interaction. PAMM **12**, 633–634 (2012)
4. Hsiao, G.C., Kleinman, R.E., Roach, G.F.: Weak solutions of fluid–solid interaction problems. Math. Nachr. **218**, 139–163 (2000)
5. Ihlenburg, F.: Finite element analysis of acoustic scattering. Springer (1998)
6. Jones, D.S.: Low–frequency scattering by a body in lubricated contact. Quart. J. Mech. Appl. Math. **36**, 111–138 (1983)
7. Junger, M., Feit, D.: Sound and Structures and their Interaction. MIT Press, Cambridge (1986)
8. Kimeswenger, A., Steinbach, O.: Symmetric BEM/FEM coupling for vibro–acoustic fluid–structure interaction problems (2013, in preparation)
9. Luke, C.J., Martin, P.A.: Fluid–solid interaction: acoustic scattering by a smooth elastic obstacle. SIAM J. Appl. Math. **55**, 904–922 (1995)
10. Of, G., Steinbach, O.: Coupled FE/BE formulations for the fluid–structure interaction. In: Domain Decomposition Methods in Science and Engineering XIX, *Lecture Notes in Computational Science and Engineering*, vol. 78, pp. 293–300. Springer, Heidelberg (2011)
11. Rjasanow, S., Steinbach, O.: The fast solution of boundary integral equations. Springer, New York (2007)
12. Schwarze, G.: Über die 1., 2. und 3. äussere Randwertaufgabe der Schwingungsgleichung. Math. Nachr. **28**, 337–363 (1965)
13. Steinbach, O.: Numerical approximation methods for elliptic boundary value problems. Finite and boundary elements. Springer, New York (2008)
14. Steinbach, O.: Boundary integral equations for Helmholtz boundary value and transmission problems. In: Direct and inverse problems in wave propagation and applications, *Radon Series on Computational and Applied Mathematics*, vol. 14, pp. 253–292. de Gruyter, Berlin (2013)
15. Steinbach, O., Unger, G.: Convergence analysis of a Galerkin boundary element method for the Dirichlet Laplacian eigenvalue problem. SIAM J. Numer. Anal. **50**, 710–728 (2012)

Optimized Schwarz Methods for Maxwell Equations with Discontinuous Coefficients

Victorita Dolean[1], Martin J. Gander[1], Erwin Veneros[1]

1 Introduction

After the development of optimized Schwarz methods for the Helmholtz equation [3, 4, 2, 12, 14], extensions to the more difficult case of Maxwell's equations were developed: for curl-curl formulations, see [1]. For first order formulations without conductivity, see [7], and with conductivity, see [5, 11]. For DG discretizations of Maxwell's equations, optimized Schwarz methods can be found in [8, 9, 6], and for scattering problems with applications, see [15, 16].

We present here optimized Schwarz methods for Maxwell's equations in heterogeneous media with discontinuous coefficients, and show that the discontinuities need to be taken into account in the transmission conditions in order to obtain effective Schwarz methods. For diffusive problems, it was shown in [10] that jumps in the coefficients can actually lead to faster iterations, when they are taken into account correctly in the transmission conditions. We show here that for the case of Maxwell's equations with jumps along the interfaces, one can obtain a non-overlapping optimized Schwarz method that converges independently of the mesh parameter; this is not possible without coefficient jumps.

2 Schwarz Methods for Maxwell's Equations

The time dependent Maxwell equations are

$$-\varepsilon\frac{\partial \mathscr{E}}{\partial t}+\nabla\times\mathscr{H}=\mathscr{J}\,,\qquad \mu\frac{\partial \mathscr{H}}{\partial t}+\nabla\times\mathscr{E}=0, \tag{1}$$

where $\mathscr{E}=(\mathscr{E}_1,\mathscr{E}_2,\mathscr{E}_3)^T$ is the electric field, $\mathscr{H}=(\mathscr{H}_1,\mathscr{H}_2,\mathscr{H}_3)^T$ is the magnetic field, ε is the *electric permittivity*, μ is the *magnetic permeability*, and $\mathscr{J}$ is the applied current density. We assume that the applied current density is divergence free, $\operatorname{div}\mathscr{J}=0$.

The time dependent Maxwell equations (1) are a system of hyperbolic partial differential equations, see for example [7]. This hyperbolic system has for any interface two incoming and two outgoing characteristics. Imposing incoming characteristics is equivalent to imposing the impedance condition

[1] Section de mathématiques, Université de Genève, 1211 Genève 4 e-mail: {victorita.dolean}{martin.gander}{erwin.veneros}@unige.ch

$$\mathscr{B}_n(\mathscr{E},\mathscr{H}) := \frac{\mathscr{E}}{Z}\times n + n\times(\mathscr{H}\times n) = s. \tag{2}$$

We consider in this paper the time-harmonic Maxwell equations,

$$-i\omega\varepsilon E + \nabla\times H = J, \qquad i\omega\mu H + \nabla\times E = 0, \quad \in \Omega, \tag{3}$$

and study the heterogeneous case where the domain Ω consists of two non-overlapping subdomains Ω_1 and Ω_2 with interface Γ, with piecewise constant parameters ε_j and μ_j in Ω_j, $j = 1,2$. A general Schwarz algorithm for this configuration is

$$\begin{cases} -i\omega\varepsilon_1 E^{1,n} + \nabla\times H^{1,n} = J & \text{in } \Omega_1,\\ i\omega\mu_1 H^{1,n} + \nabla\times E^{1,n} = 0 & \text{in } \Omega_1,\\ (\mathscr{B}_{n_1} + \mathscr{S}_1\mathscr{B}_{n_2})(E^{1,n},H^{1,n}) = (\mathscr{B}_{n_1} + \mathscr{S}_1\mathscr{B}_{n_2})(E^{2,n-1},H^{2,n-1}) & \text{on } \Gamma,\\ -i\omega\varepsilon_2 E^{2,n} + \nabla\times H^{2,n} = J & \text{in } \Omega_2,\\ i\omega\mu_2 H^{2,n} + \nabla\times E^{2,n} = 0 & \text{in } \Omega_2,\\ (\mathscr{B}_{n_2} + \mathscr{S}_2\mathscr{B}_{n_1})(E^{2,n},H^{2,n}) = (\mathscr{B}_{n_2} + \mathscr{S}_2\mathscr{B}_{n_1})(E^{1,n-1},H^{1,n-1}) & \text{on } \Gamma, \end{cases} \tag{4}$$

where $\mathscr{S}_j$, $j = 1,2$ are tangential, possibly pseudo-differential operators, and

$$\mathscr{B}_{n_j}(E^{j,n},H^{j,n}) = \frac{E^{j,n}}{Z_j}\times n_j + n_j\times(H^{j,n}\times n_j)$$

with $Z_j = \sqrt{\mu_j/\varepsilon_j}$, $j = 1,2$. Different choices of $\mathscr{S}_j$, $j = 1,2$ lead to different Schwarz methods [7].

3 The Classical Schwarz Method

The classical Schwarz method is exchanging characteristic information at the interfaces between subdomains, which means $\mathscr{S}_j = 0$. For the case of discontinuous coefficients and the domain $\Omega = \mathbb{R}^3$, with the Silver-Müller radiation condition $\lim_{r\to\infty} r\,(H\times n - E) = 0$, and the two subdomains $\Omega_1 = (-\infty,0)\times\mathbb{R}^2$, $\Omega_2 = (0,\infty)\times\mathbb{R}^2$, the classical Schwarz method does not converge in the presence of coefficient jumps:

Theorem 1. *For any* $(E^{1,0},H^{1,0}) \in (L^2(\Omega_1))^6$ *and* $(E^{2,0},H^{2,0}) \in (L^2(\Omega_2))^6$, *if* $\mu_1\varepsilon_2 \neq \mu_2\varepsilon_1$ *the classical Schwarz algorithm diverges in* $(L^2(\Omega_1))^6\times(L^2(\Omega_2))^6$.

Proof. Performing a Fourier transform in the *yz* plane with Fourier variables $k := (k_y,k_z)$, $|k| = k_y^2 + k_z^2$, we obtain after a lengthy calculation similar to the one found in [7] the convergence factor

$$\rho_{cla}(k,\omega_1,\omega_2,Z) = \max\{\rho_1(k,\omega_1,\omega_2,Z),\rho_2(k,\omega_1,\omega_2,Z)\}$$

with $\omega_1 = \omega\sqrt{\varepsilon_1\mu_1}$, $\omega_2 = \omega\sqrt{\varepsilon_2\mu_2}$, $Z = \sqrt{\frac{\mu_1\varepsilon_2}{\mu_2\varepsilon_1}}$ and

$$\rho_1(k,\omega_1,\omega_2,Z) = \left| \frac{\left(\sqrt{|k|^2-\omega_1^2} - i\omega_1 Z\right)\left(\sqrt{|k|^2-\omega_2^2} - i\omega_2/Z\right)}{\left(\sqrt{|k|^2-\omega_1^2} + i\omega_1\right)\left(\sqrt{|k|^2-\omega_2^2} + i\omega_2\right)} \right|^{\frac{1}{2}}, \quad (5)$$

$$\rho_2(k,\omega_1,\omega_2,Z) = \left| \frac{\left(\sqrt{|k|^2-\omega_1^2} - i\omega_1/Z\right)\left(\sqrt{|k|^2-\omega_2^2} - i\omega_2 Z\right)}{\left(\sqrt{|k|^2-\omega_1^2} + i\omega_1\right)\left(\sqrt{|k|^2-\omega_2^2} + i\omega_2\right)} \right|^{\frac{1}{2}}. \quad (6)$$

The condition $\mu_1\varepsilon_2 \neq \mu_2\varepsilon_1$ is equivalent to $Z \neq 1$. To show divergence, we consider 3 cases: if $\omega_1 > \omega_2$, we obtain for $|k| = \omega_1$

$$\rho_1^4(k,\omega_1,\omega_2,Z) = 1 + \frac{(\omega_1^2-\omega_2^2)(Z^2-1)}{\omega_1^2}, \quad \rho_2^4(k,\omega_1,\omega_2,Z) = 1 - \frac{(\omega_1^2-\omega_2^2)(Z^2-1)}{\omega_1^2 Z^2},$$

and hence if $Z > 1$ we have $\rho_2 > 1$, and if $Z < 1$ we have $\rho_1 > 1$. Therefore, the algorithm diverges for $\omega_1 > \omega_2$. Similarly if $\omega_1 < \omega_2$ we get for $|k| = \omega_2$

$$\rho_1^4(k,\omega_1,\omega_2,Z) = 1 - \frac{(\omega_2^2-\omega_1^2)(Z^2-1)}{\omega_2^2 Z^2} \quad \rho_2^4(k,\omega_1,\omega_2,Z) = 1 + \frac{(\omega_2^2-\omega_1^2)(Z^2-1)}{\omega_2^2},$$

and we obtain divergence as in the first case. Finally, if $\omega_1 = \omega_2$, we find

$$\rho_1(k,\omega_1,\omega_2,Z) = \rho_2(k,\omega_1,\omega_2,Z) = \left| \frac{\left(\sqrt{|k|^2-\omega_1^2} - i\omega_1 Z\right)\left(\sqrt{|k|^2-\omega_1^2} - i\omega_1/Z\right)}{\left(\sqrt{|k|^2-\omega_1^2} + i\omega_1\right)^2} \right|^{1/2}.$$

Setting now $|k| = \sqrt{2}\omega_1$, we get after some simplifications that

$$\rho_1^4 = \frac{1}{4}\frac{(Z^2+1)^2}{Z^2},$$

and $\rho_1^4 > 1$ is equivalent to

$$\rho_1^4 > 1 \iff (Z^2+1)^2 > 4Z^2 \iff (Z^2-1)^2 > 0,$$

which always holds, because by assumption $Z \neq 1$. So we also have divergence for the case $\omega_1 = \omega_2$. □

The case of continuous coefficients is analyzed in [7]. In this case, $\rho_1 = \rho_2$, and $\rho_{cla}(|k|) < 1$ for the propagative modes, $|k| < \omega_j$, $j = 1,2$, and $\rho_{cla}(|k|) = 1$ for the evanescent modes, $|k| > \omega_j$, $j = 1,2$, so the algorithm is stagnating for all evanescent modes. This is also the case if $\mu_1\varepsilon_2 = \mu_2\varepsilon_1$ which was excluded in Theorem 1 .

Having seen that the classical Schwarz method for Maxwell's equations in three dimensions diverges for most cases in the presence of coefficient jumps, we ana-

lyze now the special case of the two dimensional transverse magnetic (TMz) and transverse electric (TEz) Maxwell equations. In the TMz case, the unknowns are independent of z, and we have $E = (0,0,E_z)$ and $H = (H_x,H_y,0)$. In the TEz case, $E = (E_x,E_y,0)$ and $H = (0,0,H_z)$. Since we obtain identical results in the TMz case and the TEz case (one just has to exchange the roles of ε with μ), we only show the TMz case. Our results are again based on Fourier transforms, here in the y direction with Fourier variable k. After a similar computation as in the proof of Theorem 1, we obtain for the classical Schwarz algorithm for the TMz case the convergence factor

$$\rho_{cla}(k,\omega_1,\omega_2,Z) = \left| \frac{\left(\sqrt{k^2-\omega_1^2} - i\omega_1 Z\right)\left(\sqrt{k^2-\omega_2^2} - i\omega_2/Z\right)}{\left(\sqrt{k^2-\omega_1^2} + i\omega_1\right)\left(\sqrt{k^2-\omega_2^2} + i\omega_2\right)} \right|^{\frac{1}{2}}. \qquad (7)$$

For the TMz formulation, the classical Schwarz algorithm can be convergent in the presence of coefficient jumps:

Theorem 2. *Let $\mu_1 = \mu_2$. If $\varepsilon_1 < \varepsilon_2$ and $\sqrt{\frac{\varepsilon_1}{\varepsilon_2}} > C_0$, or if $\varepsilon_1 > \varepsilon_2$ and $\sqrt{\frac{\varepsilon_2}{\varepsilon_1}} > C_0$, $C_0 = 0.3213357548...$, then the classical Schwarz algorithm for the TMz case is convergent.*

Proof. We can only give an outline of the proof: without loss of generality, we can assume that $\omega_1 < \omega_2$. We then proceed in three steps: first, we show that for the evanescent modes, $k > \omega_j$, $j = 1,2$ we have $\rho_{cla} < 1$ if $\varepsilon_1 \neq \varepsilon_2$. Second, we show that ρ_{cla} at $k = 0$ and $k = \omega_1$ is strictly less than one, and finally we show that the maximum of those two values bounds ρ_{cla} for all the propagative modes $k < \omega_j$, $j = 1,2$, where the restriction involving C_0 comes in.

Theorem 3. *If $\varepsilon_1 = \varepsilon_2$ and $\mu_1 \neq \mu_2$, then the classical Schwarz algorithm for the TMz case is divergent.*

Proof. The proof is based on divergence of the evanescent modes, as in Theorem 1.

Theorem 4. *If $\mu_1 \neq \mu_2$, $\varepsilon_1 \neq \varepsilon_2$ and $Z < \frac{\omega_2}{\omega_1} < \frac{\sqrt{2}}{2}$, then the classical Schwarz algorithm for the TMz case is divergent.*

Proof. The proof is based again on divergence of the evanescent modes.

4 Optimized Schwarz Methods

We have seen that the classical Schwarz method is not an effective solver for Maxwell's equations in the presence of coefficient jumps. We develop now more effective transmission conditions in order to obtain optimized Schwarz methods which take the coefficient jumps into account. Using again Fourier analysis, we can show that if $\mathscr{S}_j$, $j = 1,2$ have the constant Fourier symbol

$$\widehat{\mathscr{S}_1} = -\frac{s_2 - i\omega_2 Z^{-1}}{s_2 + i\omega_2}, \quad \widehat{\mathscr{S}_2} = -\frac{s_1 - i\omega_1 Z}{s_1 + i\omega_1}, \tag{8}$$

then the optimized Schwarz method for the TMz case has the convergence factor

$$\rho_{\text{opt}}(\omega_1, \omega_2, \mu_1, \mu_2, s1, s2, k) = \left| \frac{\left(\sqrt{k^2 - \omega_1^2} - s_1\right)\left(\sqrt{k^2 - \omega_2^2} - s_2\right)}{\left(\sqrt{k^2 - \omega_1^2} + s_2 \frac{\mu_1}{\mu_2}\right)\left(\sqrt{k^2 - \omega_2^2} + s_1 \frac{\mu_2}{\mu_1}\right)} \right|^{\frac{1}{2}}. \tag{9}$$

In order to have a more efficient algorithm, we have to choose s_j, $j = 1,2$ such that ρ_{opt} is as small as possible for all numerically relevant frequencies $k \in K := [k_{\min}, k_{\max}]$, where $k_{\min}$ is a constant depending on the geometry and $k_{\max} = c_{\max}/h$, with $c_{\max}$ a constant and h denoting the mesh size, see for example [13]. We search for s_j of the form $s_j = c_j(1+i)$ such that c_j, $j = 1,2$ will be the solutions of the min-max problem

$$\rho^* = \min_{c_1, c_2 \geq 0} \left(\max_{k \in K} \rho_{\text{opt}}(\omega_1, \omega_2, \mu_1, \mu_2, k, c_1(1+i), c_2(1+i)) \right). \tag{10}$$

The proofs of the following theorems are based on asymptotic analysis, and are too long and technical for this short paper; they will appear elsewhere.

Theorem 5. *If $\mu_1 < \mu_2$ and $\frac{\mu_2}{\mu_1} > \sqrt{2}$, and $r = \sqrt{|\varepsilon_1\mu_1 - \varepsilon_2\mu_2|}$, then the asymptotic solution of the min-max problem for h small is*

$$c_1^* = \frac{1}{2} \frac{c_{max}\mu_1(\mu_2 + 2\mu_1 - \sqrt{\mu_2(4\mu_1 + 3\mu_2)})}{(2\mu_1^2 - \mu_2^2)h}, \quad c_2^* = \omega r, \tag{11}$$

$$\rho^* = \sqrt[4]{\frac{1}{2}} - \frac{2^{3/4}}{4} \frac{\omega(\mu_2^2 - 2\mu_1^2)r}{(\mu_2 + 2\mu_1 - \sqrt{\mu_2(4\mu_1 + 3\mu_2)})} h + \mathscr{O}(h^2). \tag{12}$$

If $\frac{\mu_2}{\mu_1} \leq \sqrt{2}$, then the asymptotic solution of the min-max problem is

$$c_1^* = \frac{1}{2h} \frac{c_{max}(\mu_2 - \mu_1)}{\mu_2}, \quad c_2^* = \frac{\omega r \mu_2}{2} \frac{\mu_2 + \sqrt{2\mu_1^2 - \mu_2^2}}{\mu_1^2 - \mu_2^2} \tag{13}$$

$$\rho^* = \sqrt{\frac{\mu_1}{\mu_2}} - \sqrt{\frac{\mu_1}{\mu_2}} \frac{2^{3/4}}{4} \frac{\omega r(\mu_2 + \sqrt{2\mu_1^2 - \mu_2^2})}{\mu_2 - \mu_1} h + \mathscr{O}(h^2). \tag{14}$$

Theorem 6. *If $\mu_1 = \mu_2$ and $\varepsilon_1 \neq \varepsilon_2$, and $r = \sqrt{|\varepsilon_1\mu_1 - \varepsilon_2\mu_2|}$, then the asymptotic solution of the min-max problem for h small is given by*

$$c_1^* = \left(\tfrac{c_{max}}{h}\right)^{3/4} \left(\tfrac{\omega r}{2}\right)^{1/4}, \quad c_2^* = \tfrac{1}{4} \left(\tfrac{2c_{max}}{h}\right)^{1/4} (\omega r)^{3/4}, \tag{15}$$

$$\rho^* = 1 - (\tfrac{\omega r}{2c_{max}})^{1/4} h^{1/4} + \mathscr{O}(h^{1/2}). \tag{16}$$

Theorem 7. *If $\mu_1 > \mu_2$ and $\frac{\mu_1}{\mu_2} > \sqrt{2}$, and $r = \sqrt{|\varepsilon_1\mu_1 - \varepsilon_2\mu_2|}$, then the asymptotic solution of the min-max problem for h small is*

$$c_1^* = \frac{1}{2}\frac{c_{max}\mu_2(\mu_1 + 2\mu_2 - \sqrt{\mu_1(4\mu_2+3\mu_1)})}{(2\mu_2^2 - \mu_1^2)h}, \quad c_2^* = \omega r, \tag{17}$$

$$\rho^* = \sqrt[4]{\frac{1}{2}} - \frac{2^{3/4}}{4}\frac{\omega(\mu_1^2 - 2\mu_2^2)r}{(\mu_1 + 2\mu_2 - \sqrt{\mu_1(4\mu_2+3\mu_1)})}h + \mathscr{O}(h^2), \tag{18}$$

and if $\frac{\mu_1}{\mu_2} \leq \sqrt{2}$ then the asymptotic solution of the min-max problem is

$$c_1^* = \frac{1}{2h}\frac{c_{max}(\mu_1 - \mu_2)}{\mu_1}, \quad c_2^* = \frac{\omega r\mu_1}{2}\frac{\mu_1 + \sqrt{2\mu_2^2 - \mu_1^2}}{\mu_2^2 - \mu_1^2} \tag{19}$$

$$\rho^* = \sqrt{\frac{\mu_2}{\mu_1}} - \sqrt{\frac{\mu_2}{\mu_1}}\frac{2^{3/4}}{4}\frac{\omega r(\mu_1 + \sqrt{2\mu_2^2 - \mu_1^2})}{\mu_1 - \mu_2}h + \mathscr{O}(h^2). \tag{20}$$

Theorem 5 and Theorem 7 contain the surprising result that in the presence of jumps in the coefficients, it is possible to obtain an optimized Schwarz method for Maxwell's equations with convergence factor that does not deteriorate when the mesh parameter h goes to zero, even without overlap. In the first parts of each theorem, we even see the convergence is independent of the jump in the coefficients. In the case of $\mu_1 = \mu_2$ in Theorem 6 however, the convergence factor depends on h and deteriorates as h goes to zero, as in the case in [7] when also $\varepsilon_1 = \varepsilon_2$.

5 Numerical Results

We now present some numerical results to illustrate the performance of the algorithms. We partition the domain $\Omega = (-1,1)\times(0,1)$ into two subdomains $\Omega_1 = (-1,0)\times(0,1)$ and $\Omega_2 = (0,1)^2$. In each subdomain we select constant coefficients ε_j, μ_j, $j = 1,2$. We discretize the TMz Maxwell's equations using a finite volume method, and we impose for the test on the outer boundary the impedance boundary condition $\frac{E}{Z_j}\times n_j + n_j \times (H \times n_j) = 0$, $j = 1,2$.

We first show in Figure 1 convergence histories for the classical Schwarz algorithm. On the left, we show in blue the case when $\mu_1 \neq \mu_2$ and $\varepsilon_1 = \varepsilon_2$, and in red the case when $\mu_1 \neq \mu_2$ and $\varepsilon_1 \neq \varepsilon_2$ and $Z < \frac{1}{\sqrt{2}}$, and the algorithm diverges as predicted by Theorem 3 and Theorem 4. On the right in Figure 1 we show in blue the case when $\varepsilon_1 \neq \varepsilon_2$ and $\mu_1 = \mu_2$, and in red the case when $\mu_1 \neq \mu_2$ and $\varepsilon_1 \neq \varepsilon_2$ and $Z > \frac{1}{\sqrt{2}}$, and we observe convergence, as predicted by Theorem 2.

We next show the performance of the optimized Schwarz algorithms. We call the first parts of Theorems 5 and 7 case 1, the result in Theorem 6 case 2, and the last part of Theorems 5 and 7 case 3. In Figure 2, we show scaling experiments obtained

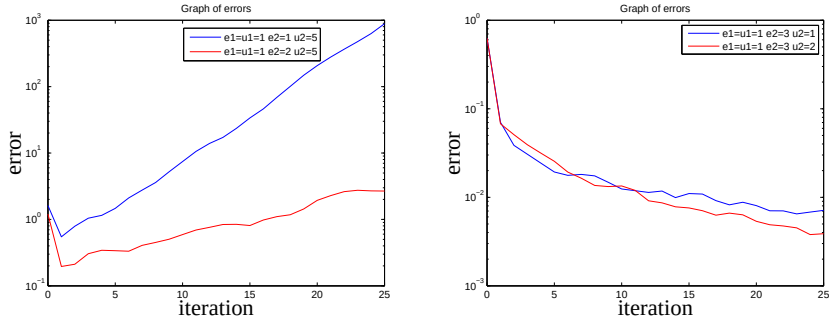
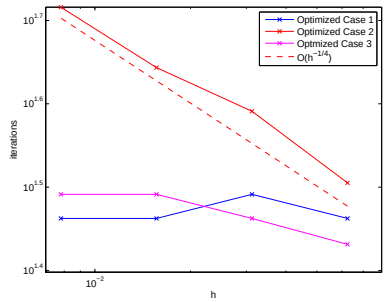


Fig. 1 Convergence histories for the classical Schwarz algorithm. On the right two cases of divergence, one where ε is continuous and one where ε is not continuous, and on the right two cases of convergence, one for μ continuous and one for μ not continuous



h	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{128}$
Optimized Case 1	29	31	29	29
Optimized Case 2	32	40	44	52
Optimized Case 3	27	29	31	31

Fig. 2 Number of iterations against the mesh size h, to attain an error of 10^{-6} with the 3 cases of the optimized Schwarz algorithm

when h is refined. Clearly case 1 and 3 lead to convergence independent of the mesh size, as predicted by Theorem 5 and Theorem 7, whereas the convergence in case 2 deteriorates, as predicted by Theorem 6. We use here the parameters $\omega = 2\pi$, $\varepsilon_1 = \mu_1 = 1$ for all the cases. For the first case we set $\varepsilon_2 = 2$ and $\mu_2 = 2$, for the second $\varepsilon_2 = 2$ and $\mu_2 = 1$ and for the third $\varepsilon_2 = 1$ and $\mu_2 = 1.4 < \sqrt{2}$.

6 Conclusions

We proved that in the presence of jumps in the coefficients, the classical Schwarz method for Maxwell's equations in 3d is not convergent, and unless $\mu_1\varepsilon_2 = \mu_2\varepsilon_1$, the algorithm actually diverges. In the 2d case of TMz and TEz modes, it is possible to obtain convergence for specific configurations of jumps. Optimized Schwarz methods on the other hand can take coefficient jumps into account and are always convergent, sometimes even better than without jumps. One can even get conver-

gence independent of the mesh parameter in the non-overlapping case, something which is impossible without coefficient jumps.

References

1. Alonso-Rodriguez, A., Gerardo-Giorda, L.: New nonoverlapping domain decomposition methods for the harmonic Maxwell system. SIAM J. Sci. Comput. **28**(1), 102–122 (2006)
2. Chevalier, P., Nataf, F.: An OO2 (Optimized Order 2) method for the Helmholtz and Maxwell equations. In: 10th International Conference on Domain Decomposition Methods in Science and in Engineering, pp. 400–407. AMS (1997)
3. Després, B.: Décomposition de domaine et problème de Helmholtz. C.R. Acad. Sci. Paris **1**(6), 313–316 (1990)
4. Després, B., Joly, P., Roberts, J.: A domain decomposition method for the harmonic Maxwell equations. In: Iterative methods in linear algebra, pp. 475–484. North-Holland, Amsterdam (1992)
5. Dolean, V., El Bouajaji, M., Gander, M.J., Lanteri, S.: Optimized Schwarz methods for Maxwell's equations with non-zero electric conductivity. In: Domain decomposition methods in science and engineering XIX, *Lect. Notes Comput. Sci. Eng.*, vol. 78, pp. 269–276. Springer, Heidelberg (2011). DOI 10.1007/978-3-642-11304-8_30. URL `http://dx.doi.org/10.1007/978-3-642-11304-8_30`
6. Dolean, V., El Bouajaji, M., Gander, M.J., Lanteri, S., Perrussel, R.: Domain decomposition methods for electromagnetic wave propagation problems in heterogeneous media and complex domains. In: Domain decomposition methods in science and engineering XIX, *Lect. Notes Comput. Sci. Eng.*, vol. 78, pp. 15–26. Springer, Heidelberg (2011). DOI 10.1007/978-3-642-11304-8_2. URL `http://dx.doi.org/10.1007/978-3-642-11304-8_2`
7. Dolean, V., Gerardo-Giorda, L., Gander, M.: Optimized Schwarz methods for Maxwell equations. SIAM J. Scient. Comp. **31**(3), 2193–?2213 (2009)
8. Dolean, V., Lanteri, S., Perrussel, R.: A domain decomposition method for solving the three-dimensional time-harmonic Maxwell equations discretized by discontinuous Galerkin methods. J. Comput. Phys. **227**(3), 2044–2072 (2008)
9. Dolean, V., Lanteri, S., Perrussel, R.: Optimized Schwarz algorithms for solving time-harmonic Maxwell's equations discretized by a discontinuous Galerkin method. IEEE. Trans. Magn. **44**(6), 954–?957 (2008)
10. Dubois, O.: Optimized Schwarz methods for the advection-diffusion equation and for problems with discontinuous coefficients. Ph.D. thesis, McGill University (2007)
11. El Bouajaji, M., Dolean, V., Gander, M.J., Lanteri, S.: Optimized Schwarz methods for the time-harmonic Maxwell equations with damping. SIAM Journal on Scientific Computing **34**(4), A2048–A2071 (2012). DOI http://dx.doi.org/10.1137/110842995
12. Gander, M., Magoulès, F., Nataf, F.: Optimized Schwarz methods without overlap for the Helmholtz equation. SIAM J. Sci. Comput. **24**(1), 38–60 (2002)
13. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**(2), 699–731 (2006)
14. Gander, M.J., Halpern, L., Magoulès, F.: An optimized Schwarz method with two-sided robin transmission conditions for the Helmholtz equation. Int. J. for Num. Meth. in Fluids **55**(2), 163–175 (2007)
15. Peng, Z., Lee, J.F.: Non-conformal domain decomposition method with second-order transmission conditions for time-harmonic electromagnetics. J. Comput. Physics **229**(16), 5615–5629 (2010). URL `http://dx.doi.org/10.1016/j.jcp.2010.03.049`
16. Peng, Z., Rawat, V., Lee, J.F.: One way domain decomposition method with second order transmission conditions for solving electromagnetic wave problems. J. Comput. Physics **229**(4), 1181–1197 (2010). URL `http://dx.doi.org/10.1016/j.jcp.2009.10.024`

Lower Dimensional Coarse Spaces for Domain Decomposition

Clark R. Dohrmann[1] and Olof B. Widlund[2]

1 Introduction

Coarse spaces are at the heart of many domain decomposition algorithms. Building on the foundation laid in [8], we have an ongoing interest in the development of coarse spaces based on energy minimization concepts [1]. Several different areas have been investigated recently, including compressible and almost compressible elasticity [3, 4], subdomains with irregular shapes [2, 11], problems in $H(curl)$ [6], and problems in $H(div)$ [12]. We also comment that there has been much recent complementary work to address problems having multiple materials in individual subdomains (see, e.g., [10, 9]).

The purpose of this study is to investigate a family of lower dimensional coarse spaces for scalar elliptic and elasticity problems. The basic idea involves the use of certain equivalence classes of nodes on subdomain boundaries. Coarse degrees of freedom are then associated with these classes, and the coarse basis functions are obtained from energy-minimizing extensions of subdomain boundary data into the subdomain interiors. We note in the context of a cube, domain decomposed into smaller cubical subdomains, that these classes are simply the subdomain vertices.

An analysis for scalar elliptic problems reveals that significant reductions in the coarse space dimension can often be achieved without sacrificing the favorable condition number estimates for larger coarse spaces. This can be important when the the memory and computational requirements associated with larger coarse spaces are prohibitive due to the use of large numbers of processors on a parallel computer. A multi-level approach could be used in such cases, but this may not always be possible or the best solution.

In the next section, we describe the nodal equivalence classes that are used in the construction of the coarse spaces. We then present algorithms for generating the coarse basis functions for different problem types in §3. An analysis for a scalar elliptic equation is provided in §4, and numerical examples are presented in §5.

[1] Computational Solid Mechanics and Structural Dynamics, Sandia National Laboratories, Albuquerque, New Mexico, 87185, USA. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000. e-mail: crdohrm@sandia.gov .[2] Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, NY 10012, USA. This work was supported in part by the National Science Foundation Grant DMS-1216564 and the U.S. Department of Energy under contracts DE-FG02-06ER25718. e-mail: widlund@cims.nyu.edu, http://cs.nyu.edu/cs/faculty/widlund/index.html

2 Coarse Nodes

Consider a domain Ω partitioned into non-overlapping subdomains $\Omega_1, \ldots, \Omega_N$. The set of all nodes common to two or more subdomains, excluding those with essential boundary conditions, is denoted by Γ_n. Let $\mathscr{S}_n$ denote the index set of subdomains containing node n. Two nodes $n_j, n_k \in \Gamma_n$ are related if $\mathscr{S}_{n_j} = \mathscr{S}_{n_k}$. As with FETI-DP or BDDC methods, we partition Γ_n into nodal equivalence classes based on this relation. Notice that for a decomposition of a cube into cubical subdomains that the nodal equivalence classes consist of faces (groups of nodes shared by the same two subdomains), edges (groups of nodes shared by the same four subdomains), and vertices (individual nodes shared by eight subdomains). For economy of words, we will henceforth use the abbreviation nec for nodal equivalence class.

Let $\mathscr{S}_{\mathscr{N}}$ denote the index set of subdomains for any node of nec $\mathscr{N}$. A nec $\mathscr{N}_j$ is said to be a child of nec $\mathscr{N}_k$ if $\mathscr{S}_{\mathscr{N}_j} \subset \mathscr{S}_{\mathscr{N}_k}$. Likewise, $\mathscr{N}_k$ is called a parent of $\mathscr{N}_j$ in this case. A nec is designated a coarse node if it is not the child of any other nec, and its coordinates are chosen as the centroid of its constituent nodes. Let $\mathscr{M}_i$ denote the set of all necs for Ω_i. Notice that each nec in $\mathscr{M}_i$ is either a coarse node or the child of at least one coarse node. Further, a coarse node c of Ω_i is also a coarse node of Ω_j for all $j \in \mathscr{S}_c$.

Notice that for the example decomposition described in the first paragraph of this section the coarse nodes are the subdomain vertices. If all necs are used as in [1], then there are approximately $(6/2 + 12/4 + 8/8)N = 7N$ necs associated with the coarse space. Likewise, if only subdomain edges and vertices are used as in [4], then there are approximately $(12/4 + 8/8)N = 4N$ necs. In contrast, the coarse space of this study is based on only about N coarse nodes.

3 Coarse Basis Functions

In this section, we describe how to construct coarse basis functions for scalar elliptic and elasticity problems in three dimensions. These coarse basis functions are fully continuous between adjacent subdomains, and we focus our attention on a single subdomain Ω_i. The support of a coarse basis function associated with coarse node c is the interior of the union of all $\bar{\Omega}_j$ with $j \in \mathscr{S}_c$.

The first step is to obtain a partition of unity for the nodes of $\Gamma_i := \partial\Omega_i \setminus \partial\Omega$. Let $\mathscr{C}_{\mathscr{N}}$ denote the set of parent coarse nodes for nec $\mathscr{N}$. If $\mathscr{N}$ is itself a coarse node, then we take $\mathscr{C}_{\mathscr{N}} = \mathscr{N}$. For the simplest case, the partition of unity associated with node $n \in \mathscr{N}$ and coarse node $c \in \mathscr{C}_{\mathscr{N}}$ is chosen as

$$p_{nc} = 1/|\mathscr{C}_{\mathscr{N}}|. \tag{1}$$

One can easily confirm that $\sum_{c \in \mathscr{C}_{\mathscr{N}}} p_{nc} = 1$.

Notice from (1) that p_{nc} is the same for all $n \in \mathscr{N}$ and $c \in \mathscr{C}_{\mathscr{N}}$. This feature can cause abrupt changes in the coarse basis functions near nec boundaries, typically

resulting in a logarithmic factor $\log(H_i/h_i)$ in estimates for the energy of the coarse basis functions. Here, H_i is the diameter of Ω_i and h_i is the diameter of its smallest element.

In an attempt to avoid the logarithmic factor, we also consider a partition of unity originating from linear functions rather than constants. Define

$$a(n) := \begin{bmatrix} 1 & x_{n1} & \cdots & x_{nd} \end{bmatrix},$$

where x_{nj} is the j-coordinate of node n and d is the spatial dimension. Let the matrix $A_{\mathcal{N}}$ denote the row concatenation of $a(n)$ for all coarse nodes in $\mathcal{C}_{\mathcal{N}}$. Notice that the number of rows of $A_{\mathcal{N}}$ is the number of parent coarse nodes for $\mathcal{N}$ and that the number of columns is $d+1$. The origin is chosen as any one of the parent coarse nodes. With reference to (1), p_{nc} is now chosen as

$$p_{nc} = a(n)A_{\mathcal{N}}^{\dagger}e_c, \tag{2}$$

where $\dagger$ denotes the Moore-Penrose pseudo-inverse and e_c is a row vector with a single nonzero entry of 1 in the row of $A_{\mathcal{N}}$ corresponding to the coarse node c. As before, one can confirm that $\sum_{c\in\mathcal{C}_{\mathcal{N}}} p_{nc} = 1$. We note if $a(n)$ is replaced by only its first column, then (2) simplifies to (1).

The energy of Ω_i is defined as $E_i(u_i) := u_i^T A_i u_i$, where u_i is a vector of nodal degrees of freedom (dofs) for Ω_i and A_i is the stiffness matrix for Ω_i. Let R_{in} select the rows of u_i for the dofs of node $n \in \mathcal{N}$. That is, $R_{in}u_i$ is the vector of dofs for node n. Let $\mathcal{N}_{ic}$ denote the set of nodes on Γ_i which have c as a parent coarse node and define

$$\Psi_{ic} := \sum_{n\in\mathcal{N}_{ic}} p_{nc}R_{in}^T N_{nc},$$

where the matrix N_{nc} is specified later for different problem types.

Let $R_{i\Gamma}$ and R_{iI} select the rows of u_i for the nodal dofs on Γ_i and the interior of Ω_i, respectively, and define

$$A_{i\Gamma\Gamma} := R_{i\Gamma}A_iR_{i\Gamma}^T, \quad A_{iI\Gamma} := R_{iI}A_iR_{i\Gamma}^T, \quad A_{iII} := R_{iI}A_iR_{iI}^T, \quad \text{etc.}$$

The coarse basis function associated with the coarse node c is given by

$$\Phi_{ic} = \Psi_{ic} - R_{iI}^T A_{iII}^{-1} A_{iI\Gamma}(R_{i\Gamma}\Psi_{ic}).$$

We note that the first term on the right hand side of this expression is the boundary data for the coarse basis function, while the second term is its energy-minimizing extension into the interior of Ω_i.

For scalar elliptic equations like the Poisson equation, we choose

$$N_{nc} = \begin{bmatrix} 1 \end{bmatrix}.$$

Remark 1. The coarse space in [2] is obtained by choosing the subdomain vertices *and* edges as the coarse nodes, and using the partition of unity given in (1). Similarly,

the smaller coarse space of [5] is obtained by choosing only the subdomain vertices as the coarse nodes and using the partition of unity given in (2).

For elasticity problems, N_{nc} is chosen as

$$N_{nc} = \begin{bmatrix} 1 & 0 & 0 & 0 & x^c_{n3} & -x^c_{n2} \\ 0 & 1 & 0 & -x^c_{n3} & 0 & x^c_{n1} \\ 0 & 0 & 1 & x^c_{n2} & -x^c_{n1} & 0 \end{bmatrix},$$

where x^c_{nj} is the j-coordinate of node n with the origin at the coarse node c. The first three columns of N_{nc} correspond to rigid body translations, while the final three columns correspond to rigid body rotations about c. We note the expression for N_{nc} can be adapted easily to accommodate finite element models with shell elements simply by adding three more rows to N_{nc}.

4 Analysis

In this section, we develop estimates for the energy of a coarse interpolant of u_i for a scalar elliptic equation. The diffusion coefficient $\rho_i > 0$ is assumed constant in Ω_i (see §4.2 of [13] for additional details). We will use the symbol u_i for both a finite element function and its vector representation in terms of nodal values. Similarly, ϕ_{ic} is the finite element function counterpart of Φ_{ic}.

For simplicity, we assume shape regular tetrahedral subdomains. In this case, the coarse basis functions for Ω_i based on (2) are identical to those for the standard P_1 linear tetrahedral element on Γ_i. Consequently, the coarse basis functions are also identical to the standard shape functions throughout Ω_i since a linear function minimizes energy for boundary data given by a linear function. We have the standard estimate

$$E_i(\phi_{ic}) \leq C H_i \rho_i. \tag{3}$$

Let $\bar{u}_i$, $\bar{u}_{\mathscr{F}}$, $\bar{u}_{\mathscr{E}}$ denote the mean of a finite element function u over the subdomain Ω_i, a subdomain face $\mathscr{F}$, and a subdomain edge $\mathscr{E}$, respectively. For a face $\mathscr{F}$ of Ω_i, it follows from the a trace theorem and a Poincaré inequality that

$$\rho_i H_i |\bar{u}_{\mathscr{F}} - \bar{u}_i|^2 \leq C E_i(u_i). \tag{4}$$

Similarly, for an edge $\mathscr{E}$ of Ω_i, we find using a discrete Sobolev inequality (see, e.g., Lemma 4.16 of [13]) that

$$\rho_i H_i |\bar{u}_{\mathscr{E}} - \bar{u}_i|^2 \leq C(1 + \log(H_i/h_i)) E_i(u_i). \tag{5}$$

Assumption 1: Let c be any vertex of Ω_i and $\mathscr{S}_c$ the index set of all subdomains containing c. Pick $j_c \in \mathscr{S}_c$ such that $\rho_{j_c} \geq \rho_j$ for all $j \in \mathscr{S}_c$. There exists a sequence $\{i = j^0_c, j^1_c, ..., j^p_c = j_c\}$ such that $\rho_i \leq C\rho_{j^\ell_c}$ and $\Omega_{j^{\ell-1}_c}$ and $\Omega_{j^\ell_c}$ have a face in common for all $\ell = 1, \ldots, p$ and $i = 1, \ldots, N$.

In other words, Assumption 1 means there is a face connected path between Ω_i and Ω_{j_c} such that the diffusion coefficient ρ_i is no greater than a constant times the diffusion coefficient of any subdomain along the path. This assumption appears to be essentially the same as the quasi-monotone assumption in [7].

Assumption 2: Using the same notation as in Assumption 1, there exists a sequence $\{i = j_c^0, j_c^1, ..., j_c^p = j_c\}$ such that $\rho_i \leq C\rho_{j_c^\ell}$ and $\Omega_{j_c^{\ell-1}}$ and $\Omega_{j_c^\ell}$ have an edge in common for all $\ell = 1, \ldots, p$ and $i = 1, \ldots, N$.

Notice that Assumption 2 is weaker than Assumption 1 since we have more options to continue at every step in the construction of a path. Our coarse interpolant of u_i for Ω_i is chosen as

$$u_{ic} = \sum_{c \in \mathscr{M}_{ic}} \bar{u}_{j_c} \phi_{ic}, \tag{6}$$

where $\mathscr{M}_{ic}$ is the set of subdomain vertices for Ω_i. Let $\mathscr{F}_{ij}$ denote the face common to Ω_i and Ω_j. Since the coarse basis functions for Ω_i can approximate constants exactly on Γ_i and also minimize the energy, it follows from a Poincaré inequality that

$$E_i(\sum_{c \in \mathscr{M}_{ic}} \bar{u}_i \phi_{ic}) \leq C E_i(u_i). \tag{7}$$

We next establish bounds for $E_i(u_{ic})$. Starting with

$$\bar{u}_i - \bar{u}_{j_c} = (\bar{u}_i - \bar{u}_{\mathscr{F}_{j_c^0 j_c^1}}) + \sum_{\ell=1}^{p-1} (\bar{u}_{\mathscr{F}_{j_c^{\ell-1} j_c^\ell}} - \bar{u}_{\mathscr{F}_{j_c^\ell j_c^{\ell+1}}}) + (\bar{u}_{\mathscr{F}_{j_c^{p-1} j_c^p}} - \bar{u}_{j_c}),$$

rewriting the term in the summation as

$$\bar{u}_{\mathscr{F}_{j_c^{\ell-1} j_c^\ell}} - \bar{u}_{\mathscr{F}_{j_c^\ell j_c^{\ell+1}}} = (\bar{u}_{\mathscr{F}_{j_c^\ell j_c^{\ell-1}}} - \bar{u}_{j_c^\ell}) - (\bar{u}_{\mathscr{F}_{j_c^\ell j_c^{\ell+1}}} - \bar{u}_{j_c^\ell}),$$

and using Assumption 1 and (4), we find

$$\rho_i H_i |\bar{u}_i - \bar{u}_{j_c}|^2 \leq C \sum_{j \in \mathscr{S}_c} E_j(u_j).$$

It then follows from (3) that

$$E_i((\bar{u}_i - \bar{u}_{j_c})\phi_{ic}) \leq C \sum_{j \in \mathscr{S}_c} E_j(u_j).$$

Finally, from (6), (7), and the triangle inequality, we obtain

$$E_i(u_{ic}) \leq C \sum_{j \in \mathscr{M}_i} E_j(u_j),$$

where $\mathscr{M}_i$ is the index set of all subdomains adjacent to Ω_i. Summing contributions from all subdomains and noting that $|\mathscr{M}_i| < C$, we see that the energy of our coarse interpolant is uniformly bounded by the energy of u. That is, under Assumption 1,

$$\sum_{i=1}^{N} E_i(u_{ic}) \leq C \sum_{i=1}^{N} E_i(u_i). \tag{8}$$

By using (5) instead of (4) in the previous development, we find under the less restrictive Assumption 2 that

$$\sum_{i=1}^{N} E_i(u_{ic}) \leq C(1+\log(H/h)) \sum_{i=1}^{N} E_i(u_i), \tag{9}$$

where $H/h := \max_i(H_i/h_i)$.

If the coarse basis functions originate from (1) rather than (2), then it follows from elementary estimates and Lemma 4.25 of [13] that an additional factor of $1+\log(H_i/h_i)$ will appear on the right-hand-side of (3). Thus, this additional factor will also be present in (8) and (9). The same also holds for hexahedral subdomains even when (2) is used since a linear function cannot interpolate a function at all four nodes of a quadrilateral planar face.

With the estimates for our coarse interpolants in hand, we may now perform a local analysis for an overlapping additive Schwarz algorithm using basically the same approach as in [2] or [5]. This involves a partition of unity $\{\vartheta_i\}_{i=1}^N$ with $0 \leq \vartheta_i \leq 1$, $|\nabla \vartheta_i| \leq C/\delta_i$, and ϑ_i supported in the closure of the overlapping subdomain Ω_i'. Here, δ_i is the thickness of the part of Ω_i' which is common to its neighbors. Given an estimate of the form

$$\sum_{i=1}^{N} E_i(u_{ic}) \leq C f(H/h) \sum_{i=1}^{N} E_i(u_i),$$

the resulting condition number estimate for the preconditioned operator is given by

$$\kappa(M^{-1}A) \leq C f(H/h)(1+H/\delta), \tag{10}$$

where $H/\delta := \max_i H_i/\delta_i$. Comparing (10) with (8) and (9), we that $f(H/h)$ is 1 and $1+\log(H/h)$ under Assumptions 1 and 2, respectively.

5 Numerical Examples

We consider a unit cube domain decomposed into either smaller cubical subdomains or irregular-shaped subdomains obtained from a mesh partitioner for a scalar elliptic equation; an analysis and results for elasticity will appear in a forthcoming study. The numbers of iterations and condition number estimates from the conjugate gradient algorithm appear under the headings *iter* and *cond* in the tables. All results are for homogeneous essential boundary conditions on one face of the cube, a random right-hand-side vector, and a relative residual solver tolerance of 10^{-8}.

The results in Table 1 are for 64 cubical subdomains and a fixed dimensionless overlap H/δ. By plotting condition numbers versus $\log(H/h)$, it appears that the

line segment slopes are bounded above by constants as H/h increases for both the constant and checkerboard material properties. Moreover, these line segment slopes for constant material properties and the linear partition of unity in (2) appear to decrease with increasing H/h, while those for (1) appear to approach a constant value. These observations are consistent with the analysis. We note for a vertex coarse space, as used in this example, a much less favorable condition number estimate of $C(H/h)(1+\log(H/h))^2$ holds for FETI-DP and BDDC algorithms (cf. Algorithm A in §6.4.2 of [13]).

Table 1 Results for constant and checkerboard arrangements of subdomain material properties ($\rho_i = 1$ or $\rho_i = 10^4$) for partitions of unity based on (1) and (2). The overlap $H/\delta \approx 4$ is held fixed while H/h varies.

	constant				checkerboard			
	p_{nc} (1)		p_{nc} (2)		p_{nc} (1)		p_{nc} (2)	
H/h	iter	cond	iter	cond	iter	cond	iter	cond
8	40	29.0	37	25.2	37	39.9	35	29.7
12	43	33.3	38	27.7	40	46.4	37	32.5
16	45	36.4	39	29.3	40	50.9	38	34.4
20	45	38.8	39	30.5	41	54.1	38	35.7

For the final example, we consider a mesh of 48^3 elements decomposed into different numbers of subdomains using a mesh partitioner. Results in Table 2 show that the present coarse space dimensions are significantly smaller than those for the richer coarse space in [1]. Smaller dimensional coarse spaces result in reduced computational requirements for the coarse problem, and extend the range of problem sizes that can be solved effectively using a two-level method.

Table 2 Results for constant coefficients and a mesh with 48^3 elements decomposed using a mesh partitioner. The coarse space dimension is denoted by n_c and the overlap is for two layers of additional elements. The final row in the table is for a regular mesh decomposition into 64 identical subdomains.

	Ref. [1]			p_{nc} (1)			p_{nc} (2)		
N	n_c	iter	cond	n_c	iter	cond	n_c	iter	cond
63	831	45	21.3	166	46	22.5	166	40	15.7
64	863	45	21.5	174	46	22.5	174	41	16.4
65	916	46	21.1	189	46	21.7	189	40	16.6
$64^\star$	279	40	24.9	27	43	33.3	27	38	27.7

References

1. Dohrmann, C.R., Klawonn, A., Widlund, O.B.: A family of energy minimizing coarse spaces for overlapping Schwarz preconditioners. In: U. Langer, M. Discacciati, D. Keyes, O. Widlund, W. Zulehner (eds.) Proceedings of the 17th International Conference on Domain Decomposition Methods in Science and Engineering, held in Strobl, Austria, July 3-7, 2006, no. 60 in Springer-Verlag, Lecture Notes in Computational Science and Engineering, pp. 247–254 (2007)
2. Dohrmann, C.R., Klawonn, A., Widlund, O.B.: Domain decomposition for less regular subdomains: Overlapping Schwarz in two dimensions. SIAM J. Numer. Anal. **46**(4), 2153–2168 (2008)
3. Dohrmann, C.R., Widlund, O.B.: An overlapping Schwarz algorithm for almost incompressible elasticity. SIAM J. Numer. Anal. **47**(4), 2897–2923 (2009)
4. Dohrmann, C.R., Widlund, O.B.: Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity. Internat. J. Numer. Meth. Engng. **82**, 157–183 (2010)
5. Dohrmann, C.R., Widlund, O.B.: An alternative coarse space for irregular subdomains and an overlapping Schwarz algorithm for scalar elliptic problems in the plane. SIAM J. Numer. Anal. **50**(5), 2522–2537 (2012)
6. Dohrmann, C.R., Widlund, O.B.: An iterative substructuring algorithm for two-dimensional problems in H(curl). SIAM J. Numer. Anal. **50**(3), 1004–1028 (2012)
7. Dryja, M., Sarkis, M.V., Widlund, O.B.: Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. Numer. Math. **72**, 313–348 (1996)
8. Dryja, M., Smith, B.F., Widlund, O.B.: Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. SIAM J. Numer. Anal. **31**(6), 1662–1694 (1994)
9. Galvis, J., Efendiev, Y.: Domain decomposition preconditioners for multiscale flows in high-contrast media. Multiscale Model. Simul. **8**(4), 1461–1483 (2010)
10. Graham, I.G., Lechner, P.O., Scheichl, R.: Domain decomposition for multiscale PDEs. Numer. Math. **106**(4), 589–626 (2007)
11. Klawonn, A., Rheinbach, O., Widlund, O.B.: An analysis of a FETI–DP algorithm on irregular subdomains in the plane. SIAM J. Numer. Anal. **46**(5), 2484–2504 (2008)
12. Oh, D.S.: An overlapping Schwarz algorithm for Raviart-Thomas vector fields with discontinuous coefficients. SIAM J. Numer. Anal. **51**(1), 297–321 (2013)
13. Toselli, A., Widlund, O.: Domain Decomposition Methods - Algorithms and Theory, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin Heidelberg New York (2005)

Robust Preconditioners for DG-Discretizations with Arbitrary Polynomial Degrees

Kolja Brix[1], Claudio Canuto[2], and Wolfgang Dahmen[1]

1 Introduction

Discontinuous Galerkin (DG) methods offer an enormous flexibility regarding local grid refinement and variation of polynomial degrees rendering such concepts powerful discretization tools which have proven to be well-suited for a variety of different problem classes. While initially the main focus has been on transport problems like hyperbolic conservation laws, interest has meanwhile shifted towards diffusion problems. Specifically, we focus here on the efficient solution of the linear systems of equations that arise from the Symmetric Interior Penalty DG method applied to elliptic boundary value problems. [1] The principal objective is to develop robust preconditioners for the full "DG-flexibility" which means to obtain uniformly bounded condition numbers for locally refined meshes and arbitrarily (subject to mild grading conditions) varying polynomial degrees at the expense of linearly scaling computational work. A first step towards that goal has been made in [3] treating the case of geometrically conforming meshes but arbitrarily large variable polynomial degrees which already exposes major principal obstructions. In this paper we complement this work by detailed studies of several issues arising in [3].

To our knowledge the only concept yielding full robustness with respect to polynomial degrees is based on *Legendre-Gauß-Lobatto* (LGL) quadrature. Specifically, in the framework of *auxiliary space methods* low order finite element discretizations on LGL-grids can be used to precondition high order polynomial discretizations. However, when dealing with variable degrees the possible non-matching of such grids at element interfaces turns out to severely obstruct in general the design of efficient preconditioners. To overcome these difficulties we propose in [3] a concatenation of auxiliary space preconditioners. In the first stage the spectral DG formulation (**SE-DG**) is transferred to a spectral continuous Galerkin formulation (**SE-CG**). In the second stage we proceed from here to a finite element formulation on a specific dyadic grid (**DFE-CG**) which is associated with an LGL-grid but belongs to a nested hierarchy. The latter problem can then be tackled by a multilevel wavelet preconditioner presented in forthcoming work. The overall path of our iterated auxiliary space preconditioner therfore is **SE-DG** $\rightarrow$ **SE-CG** $\rightarrow$ **DFE-CG**. It should be noted that a natural alternative is to combine the first stage with a domain decomposition substructuring preconditioner as proposed in [6] admitting a mild growth of condition numbers with respect to the polynomial degree.

[1] Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, 52056 Aachen, Germany, e-mail: {brix}{dahmen}@igpm.rwth-aachen.de .[2] Dipartimento di Scienze Matematiche, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy, e-mail: claudio.canuto@polito.it

We are content here for most part of the paper with brief pointers to the detailed analysis in [3], [4] and [2] to an extent needed for the present discussion.

Section 2 introduces our model problem, the LGL technique is explained in Section 3. The auxiliary space method is detailed in Section 4, while Sections 5 and 6 consider stages 1 and 2 of our preconditioner. Finally in Section 7 we give some numerical experiments that shed light on the constants that arise in four basic inequalities used in the second stage.

2 Model problem and Discontinuous Galerkin formulation

Given a bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$ with piecewise smooth boundary we consider as a simple model problem the weak formulation: find $u \in H_0^1(\Omega)$ such that

$$a(u,v) := \int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x = \langle f, v \rangle, \quad v \in H_0^1(\Omega)$$

of Poisson's equation $-\Delta u = f$ on Ω with zero Dirichlet boundary conditions $u = 0$ on $\partial\Omega$. For simplicity, we assume that $\bar{\Omega}$ is the union of a collection $\mathscr{R}$ of finitely many (hyper-)rectangles, which at most overlap with their boundaries. More complex geometries can be handled by isoparametric mappings. By $\mathscr{F}_l(R)$ we denote the l-dimensional facets of a (hyper-)rectangle R and by $\mathscr{F}_l = \cup_{R\in\mathscr{R}} \mathscr{F}_l(R)$ the union of all these objects. Let $H_k(R)$ be the side length of R in the k-th coordinate direction.

The polynomial degrees used in each cell R are defined as $p = (p_k)_{k=1}^d$, where p_k is the polynomial degree in the k-th coordinate direction. We introduce the piecewise constant function $\delta = (H, p)$ that collects the hp approximation parameters. On δ we impose mild grading conditions, see [3] for the details.

For the spectral discretization of our model problem, we choose the DG spectral ansatz space $V_\delta := \{v \in L^2(\Omega) : v|_R \in \mathbb{Q}_p(R) \text{ for all } R \in \mathscr{R}\}$, where $\mathbb{Q}_p(R)$ is the tensor space of all polynomials of degree at most p on the (hyper-)rectangle R.

We employ the standard notation of DG methods for jumps and averages on the mesh skeleton and on $\partial\Omega$. The *Symmetric Interior Penalty Discontinuous Galerkin* method (SIPG) $a_\delta(u,v) = \langle f, v\rangle$ for all $v \in V_\delta$ with the SIPG bilinear form

$$\begin{aligned} a_\delta(u_\delta, v_\delta) := \sum_{R\in\mathscr{R}} (\nabla u_\delta, \nabla v_\delta)_R + \sum_{F\in\mathscr{F}} (-(\{\nabla u_\delta\}, [v_\delta])_F - ([u_\delta], \{\nabla v_\delta\})_F) \\ + \sum_{F\in\mathscr{F}} \gamma \omega_F ([u_\delta], [v_\delta])_F = (f, v_\delta)_\Omega, \quad v_\delta \in V_\delta \end{aligned}$$

with $\omega_F := \max\{\omega_{F,R^-}, \omega_{F,R^+}\}$ for internal faces F and $\omega_{F,R^\pm} := \frac{p_k(R^\pm)(p_k(R^\pm)+1)}{H_k(R^\pm)}$. For boundary faces $F \subset \partial\Omega$ we set $\omega_{F,R} := \frac{p_k(R)(p_k(R)+1)}{H_k(R)}$.

3 Legendre-Gauß-Lobatto (LGL) grids

Denoting by $(\xi_i)_{i=1}^{p-1}$ the zeros of the first derivative of the p-th Legendre polynomial L_p, (in ascending order), and setting $\xi_0 = -1$ and $\xi_p = 1$, $\mathscr{G}_p = (\xi_i)_{0\le i\le p}$ is the Legendre-Gauß-Lobatto (LGL) grid of degree p on the reference interval $\hat{I} = [-1,1]$, see e.g. [5]. In combination with appropriate LGL weights $(w_i)_{0\le i\le p}$ the LGL points of order p can be interpreted as quadrature points of a quadrature rule of exactness $2p-1$. In [4] we prove quasi-uniformity of the LGL-grids $(\mathscr{G}_p)_{p\in\mathbb{N}}$, i.e., $\frac{h_{i+1,p}}{h_{i,p}}$ remains uniformly bounded independent of p, where $h_i = |\xi_i - \xi_{i-1}|$ for $1 \le i \le p-1$.

The particular relevance of tensor product LGL-grids for preconditioners for spectral element discretizations lies in the two norm equivalences (see [5])

$$\|\varphi\|_{H^i(R)} \eqsim \|\mathscr{I}_{h,p}^R \varphi\|_{H^i(R)} \quad \text{for all} \quad \varphi \in \mathbb{Q}_p(R), \quad i \in \{0,1\}, \tag{1}$$

which hold uniformly for any d-dimensional hypercube $R = \times_{k=1}^d I_k$ where $\mathscr{I}_{h,p}^R$ is the piecewise multi-linear interpolant on the tensor product LGL-grid.

4 Abstract theory: Auxiliary Space Method

The auxiliary space method (ASM) [9, 11, 10] is a powerful concept for the construction of preconditioners that can be derived from the *fictitious space lemma* [8, 7, 9].

Given a problem $a(u,v) = f(v)$ for all $v \in V$ on the linear space V equipped with a bilinear form $a(\cdot,\cdot) : V \times V \to \mathbb{R}$, we seek an *auxiliary space* $\tilde{V}$ with an *auxiliary form* $\tilde{a}(\cdot,\cdot) : \tilde{V} \times \tilde{V} \to \mathbb{R}$ that is in some sense close to the original one but easier to solve. Note that we neither require $V \subset \tilde{V}$ nor $\tilde{V} \subset V$ which is important in the context of non-conforming discretizations. Therefore on the sum $\hat{V} = V + \tilde{V}$ we need in general another version $\hat{a}(\cdot,\cdot) : \hat{V} \times \hat{V} \to \mathbb{R}$ as well as a second form $b(\cdot,\cdot) : \hat{V} \times \hat{V} \to \mathbb{R}$ which dominates a on V and plays the role of a smoother. The required closeness of the spaces V and $\tilde{V}$ is described with the aid of two linear operators $Q : \tilde{V} \to V$ and $\tilde{Q} : V \to \tilde{V}$. Specifically, these operators have to satisfy certain direct estimates involving the above bilinear forms. For the details on the ASM conditions see [9].

Lemma 1 (Stable Splitting [9]). *Under the assumptions of the ASM, we have the following stable splitting*

$$a(v,v) \sim \inf_{w\in V,\tilde{v}\in\tilde{V}:\ v=w+Q\tilde{v}} (b(w,w) + \tilde{a}(\tilde{v},\tilde{v})) \quad \textit{for all } v \in V.$$

The main result of the ASM is given in the following theorem [9].

Theorem 1 (Auxiliary Space Method). *Let* $\mathbf{C_B}$ *and* $\mathbf{C_{\tilde{A}}}$ *be symmetric preconditioners for* $\mathbf{B}$ *and* $\tilde{\mathbf{A}}$*, respectively. Let* $\mathbf{S}$ *be the representation of* $Q : \tilde{V} \to V$*. Then* $\mathbf{C_A} := \mathbf{C_B} + \mathbf{S}\mathbf{C_{\tilde{A}}}\mathbf{S}^T$ *is a symmetric preconditioner for* $\mathbf{A}$*. Moreover, there exists a*

uniform constant C such that the spectral condition number of $\mathbf{C_A A}$ satisfies

$$\kappa(\mathbf{C_A A}) \leq C\kappa(\mathbf{C_B B})\kappa(\mathbf{C_{\tilde{A}}\tilde{A}}).$$

For a given practical application it remains to identify a suitable auxiliary space $\tilde{V}$, the bilinear forms $\tilde{a}:\tilde{V}\times\tilde{V}\to\mathbb{R}$ and $\hat{a},b:\hat{V}\times\hat{V}\to\mathbb{R}$, as well as the two linear operators Q and $\tilde{Q}$, such that ASM conditions are satisfied. In addition efficient preconditioners for the "easier" auxiliary problems $\mathbf{C_{\tilde{A}}}$ and $\mathbf{C_B}$ need to be devised. Of course, the rationale is that the complexity to apply $\mathbf{C_{\tilde{A}}}$ and $\mathbf{C_B}$ should be much lower than solving the original problem.

Note that the operator $\tilde{Q}$ need *not* be implemented but enters only the analysis.

5 Stage 1: ASM DG-SEM $\to$ CG-SEM

In the first stage, we choose the largest conforming subspace $\tilde{V} := V_\delta \cap H^1_0(\Omega)$ of $V := V_\delta$ as auxiliary space so that Q can be taken as the canonical injection. The definition of the operator $\tilde{Q}$ can be found in [3].

The main issue in this stage is the choice of the auxiliary form $b(\cdot,\cdot)$. Using LGL-quadrature combined with an inverse estimate for the partial derivatives in the form $a(\cdot,\cdot)$ we arrive at

$$b(u,v) := \sum_{R\in\mathscr{R}}\sum_{\xi\in\mathscr{G}_p(R)} u(\xi)\,v(\xi)\,c_\xi W_\xi, \quad W_\xi := \left(\sum_{k=1}^{d} w_{\xi,k}^{-2}\right) w_{\xi,k}.$$

Here the weights $c_\xi \sim 1$ are chosen as

$$c_\xi := \begin{cases} \beta_1(c_1^2 + \gamma\rho_1\omega_F w_{F,R}/W_\xi), & \xi\in\mathscr{G}_p(F,R),\ F\in\mathscr{F}_{d-1}(R),\ R\in\mathscr{R}, \\ \beta_1 c_1^2, & \text{else}, \end{cases}$$

where $w_{F,R^\pm}$ is the LGL quadrature weight on F seen as a face of $R^\pm$ and the parameters β,ρ_1 can be used to "tune" the scheme. The resulting matrix $\mathbf{B}$ is diagonal so that the application of $\mathbf{C_B} := \mathbf{B}^{-1}$ requires only $\mathscr{O}(N)$ operations. It is shown in [3] that all ASM conditions are satisfied for this choice of $b(\cdot,\cdot)$. Numerical experiments show that the parameters β_1 and ρ_1 can by and large be optimized independently of the polynomial degrees.

6 Stage 2: CG-SEM $\to$ CG-DFEM

The second stage involves three major ingredients, namely

(1) the choice of spaces of piecewise multi-linear finite elements on hierarchies of *nested* anisotropic dyadic grids, to permit a subsequent application of efficient multilevel preconditioners,
(2) the construction of the operators Q and $\tilde{Q}$, and
(3) the construction of the auxiliary bilinear form $b(\cdot,\cdot)$.

As for (1), the non-matching of LGL-grids for different degrees p at interfaces prevents us from taking low order finite element spaces as auxiliary space for the high order conforming problem resulting from the first stage. Instead, with each LGL-grid $\mathscr{G}_p$ we associate a dyadic grid $\mathscr{G}_{D,p}$, which is roughly generated as follows: starting with the boundary points $\{-1,1\}$ as initial guess we adaptively refine the grid. A subinterval in the grid is bisected into two parts of equal size, if the smallest of the overlapping LGL-subintervals is longer than α times its length. The parameter α therefore controls the mesh size of the dyadic grid. However, for input LGL-grids of different polynomial degrees the resulting dyadic grids are not necessarily nested yet. How to ensure nestedness while keeping the grid size under control is shown in [3]. The key quality of the associated dyadic grids $\mathscr{G}_{D,p}$ is that mutual low order piecewise multi-linear interpolation between the low order finite element spaces on $\mathscr{G}_p(R), \mathscr{G}_{D,p}(R)$ is uniformly H^1-stable, see [3] for the proofs. Denoting by $V_{h,D,p}(R)$ the space of piecewise multi-linear conforming finite elements on $\mathscr{G}_{D,p}(R)$, we now take $V := V_\delta \cap H^1_0(\Omega)$ and $\tilde{V} := V_{h,D} \cap H^1_0(\Omega)$, where $V_{h,D} = \{v \in C^0(\overline{\Omega}) \,:\, \forall R \in \mathscr{R}\ ,\ v_{|R} := v_R \in V_{h,D,p}(R)\}$.

Concerning (2), the operator Q is defined element-wise as follows. For a given element vertex $z \in \mathscr{F}_0(R)$ let p^* denote the polynomial degree vector whose kth entry is the minimum of the kth entries of all degree vectors associated with elements R' sharing z as a vertex. Here a grading of the degrees is important. Let $\Phi_z \in \mathbb{Q}_1(R)$ the multi-linear shape function on R satisfying conditions $\Phi_z(y) = \delta_{y,z}$ for all $y \in \mathscr{F}_0(R)$. Then, we define

$$\tilde{v}^*_z := \mathscr{I}^R_{h,D,p^*_z}(\Phi_z \tilde{v}_R) \in V_{h,D,p^*_z}(R) \qquad \text{and} \qquad v^*_z = \mathscr{I}^R_{p^*_z} \tilde{v}^*_z \in \mathbb{Q}_{p^*_z}(R)\,, \tag{2}$$

where $\mathscr{I}^R_{h,D,p^*_z}, \mathscr{I}^R_{p^*_z}$ are the dyadic piecewise multilinear and high order LGL-interpolants on the respective grids. Summing-up over the vertices of R, we define

$$\tilde{v}^*_R := \sum_{z \in \mathscr{F}_0(R)} \tilde{v}^*_z \in V_{h,D,p}(R) \qquad \text{and} \qquad Q_R \tilde{v}_R := v^*_R := \sum_{z \in \mathscr{F}_0(R)} v^*_z \in \mathbb{Q}_p(R)\,. \tag{3}$$

The operator $\tilde{Q}$ is defined analogously with the roles of dyadic and LGL-grids exchanged, see [3].

To finally address (3), for the structure of the form $b(\cdot,\cdot)$ from the first stage the direct estimates in the ASM conditions are no longer valid. It has to be suitably relaxed along the following lines. We make an ansatz of the form

$$b(v,w) := \sum_{R \in \mathscr{R}} \sum_{k=1}^{d} \Big(\sum_{S_\ell \in \mathscr{T}_{0,k}(R)} b^0_{R,k,S_\ell}(v,w) + \sum_{S_\ell \in \mathscr{T}_{1,k}(R)} b^1_{R,k,S_\ell}(v,w) \Big), \tag{4}$$

where $\mathscr{T}_{0,k}(R)$ is the collection of those LGL-subcells S_ℓ, $\ell \in \times_{k=1}^d \{1,\dots,p_k(R)\}$ with side lengths $h_l^{(\ell_l)}$ in the LGL-grid $\mathscr{G}_p(R)$ that are *strongly anisotropic* according to $(\max_{l \neq k} h_l^{(\ell_l)})/h_k^{(\ell_k)} > C_{\text{aspect}}$ for a fixed constant $C_{\text{aspect}} > 0$, while $\mathscr{T}_{1,k}(R)$ is comprised of the remaining "isotropic" cells. On the isotropic cells in $\mathscr{T}_{1,k}(R)$ we

use an inverse estimate applied to piecewise multi-linear LGL-interpolants of v and w. On the remaining anisotropic cells we retain integrals over the variable involving the partial derivative and use quadrature in the remaining variables. For this auxiliary form $b(\cdot,\cdot)$ and the above operators Q and $\tilde{Q}$ we can verify all ASM conditions, see [3]. Note that the Gramian $\mathbf{B}$ is no longer diagonal and we refer to [3] for efficient realizations of $\mathbf{C_B}$.

7 Numerical experiments: Constants in the basic interpolation inequalities

A fundamental role in the proof of the ASM-conditions in the second stage **SE-CG** $\rightarrow$ **DFE-CG** is played by four basic interpolation estimates. In particular, knowing the size of the constants arising in these inequalities and their dependence on the polynomial degrees helps understanding the quantitative effects observed in more complex situations later on.

As before, let Φ_z denote the affine shape function now on the reference interval $\hat{I} = [-1,1] \subset \mathbb{R}$ satisfying $\Phi_z(x) = \delta_{x,z}$ for $x,z \in \{-1,1\}$. By $\mathscr{I}_q$ we denote the polynomial interpolation operator on the LGL-grid $\mathscr{G}_q$ for polynomial degree q and by $\mathscr{I}_{h,D,q}$ the piecewise affine interpolation operator on the dyadic grid $\mathscr{G}_{D,q}$ associated with $\mathscr{G}_q$.

A major tool for proving the ASM conditions is given by the following theorem.

Theorem 2. *Assume that $cp \le q \le p$ for some fixed constant $c > 0$. Then we have*

$$|\mathscr{I}_q(\Phi_z v)|_{H^m(\hat{I})} \lesssim \|v\|_{H^m(\hat{I})} \quad \textit{for all } v \in \mathbb{Q}_p(\hat{I}),\ z \in \{-1,1\},\ m \in \{0,1\}, \tag{5}$$

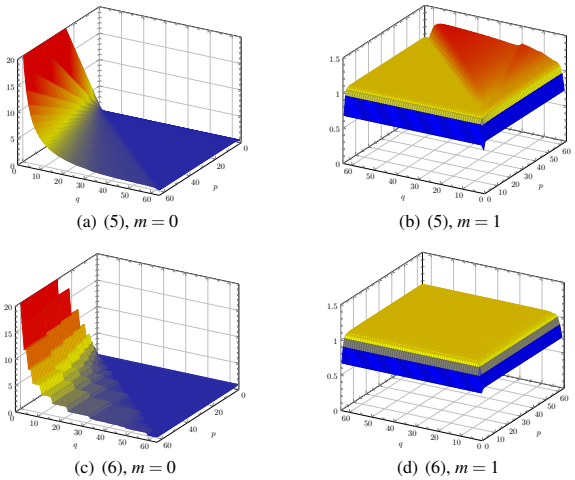
and

$$|\mathscr{I}_{h,D,q}(\Phi_z \tilde{v})|_{H^m(\hat{I})} \lesssim \|\tilde{v}\|_{H^m(\hat{I})} \quad \textit{for all } \tilde{v} \in V_{h,D,p}(\hat{I}),\ z \in \{-1,1\},\ m \in \{0,1\}. \tag{6}$$

We determine next *numerically* the smallest constants that fulfill the inequalities (5) and (6). This can be obtained by solving generalized eigenvalue problems for the largest generalized eigenvalue. For all dyadic grids we choose the grid generation parameter $\alpha = 1.2$, which balances two effects: on the one hand, the generated auxiliary space is rich enough for a good approximation while on the other hand, to keep the solution of the auxiliary space feasible, the dyadic grid does not have too many degrees of freedom. Figure 1 shows the dependence of the smallest possible constants on the polynomial degrees p and q in the range $1 \le p,q \le 64$.

We observe that the constants in (5) and (6) become large for $m = 0$ when the quotient p/q increases, but eventually stay bounded as long as $cp \le q \le p$ for a fixed $c > 0$. For $m = 1$ we find uniform moderate constants in (5) and (6) for arbitrary choices of p and q. While the nodes in the LGL-grids move gradually with increasing degree the associated dyadic grids change more abruptly which explains the jumps in the graph in Figure 1(c).

We are particularly interested in the behavior of the constants when the quotient of p and q is fixed, i.e., we restrict ourselves to a cross section through the

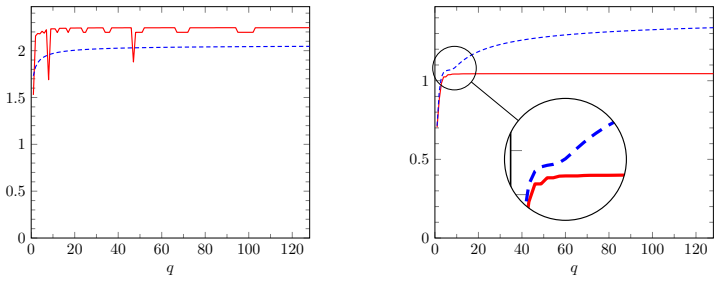


(a) (5), $m = 0$ (b) (5), $m = 1$

(c) (6), $m = 0$ (d) (6), $m = 1$

Fig. 1 Dependance of the constants in (5) and (6) on p and q.

3-dimensional plots along a line in the pq-plane. As an example, we choose $p = 2q$ representing strongly varying degrees on adjacent elements. The smallest constants in the inequalities for polynomial degrees q up to 128 are displayed in Figure 2.



(a) m=0 (b) m=1

Fig. 2 Constants in the basic interpolation inequalities for $p = 2q$ (dashed line: (5), solid line: (6)).

While for $m = 0$ the constants quickly approach an asymptotic value for both (5) and for (6), this is not true for (5) and $m = 1$. In this case we observe a very slow monotonic convergence to its asymptotic limit. Thus for moderate polynomial degrees one still observes a significant growth. Since this estimate is relevant for the ASM conditions on the operator $\tilde{Q}$ in the second stage, this leads to some growth of the condition number of the preconditioned problem for moderate polynomial degrees and significant inter-element jumps, although it eventually stays uniformly bounded independent of the polynomial degree q.

8 Summary and outlook

In this paper we sketch a preconditioner for the spectral symmetric interior penalty discontinuous Galerkin method that, under mild grading conditions, is robust in variably arbitrarily large polynomial degrees, announcing detailed results given in [3]. The concept is based on the LGL-techniques for spectral methods combined with judiciously chosen nested dyadic grids through an iterated application of the auxiliary space method. A detailed exposition of a multiwavelet preconditioner for the dyadic grid problem, an extension to locally refined grids with hanging nodes, strategies for parallel implementations, and the treatment of jumping coefficients will be presented in forthcoming work.

Acknowledgements We thank for the support by the Seed Funds project funded by the Excellence Initiative of the German federal and state governments and by the DFG project 'Optimal preconditioners of spectral Discontinuous Galerkin methods for elliptic boundary value problems' (DA 117/23-1).

References

1. Ayuso De Dios, B.: Solvers for discontinuous Galerkin methods. In: Proceedings of the 21st International Conference on Domain Decomposition Methods, Rennes, June 2012 (2012)
2. Brix, K.: Robust preconditioners for hp-discontinuous Galerkin discretizations for elliptic problems. Ph.D. thesis, Institut für Geometrie und Praktische Mathematik, RWTH Aachen, in preparation.
3. Brix, K., Campos Pinto, M., Canuto, C., Dahmen, W.: Multilevel preconditioning of discontinuous Galerkin spectral element methods. Part I: Geometrically conforming meshes. IGPM Preprint #355, RWTH Aachen. (2013). Submitted. `arXiv:1301.6768 [math.NA]`
4. Brix, K., Canuto, C., Dahmen, W.: Legendre-Gauss-Lobatto grids and associated nested dyadic grids. IGPM Preprint #378, RWTH Aachen. (2013). Submitted. `arXiv:1311.0028 [math.NA]`
5. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: Spectral methods. Fundamentals in single domains. Springer, Berlin (2006)
6. Canuto, C., Pavarino, L.F., Pieri, A.B.: BDDC preconditioners for continuous and discontinuous Galerkin methods using spectral/hp elements with variable polynomial degree. IMA J. Numer. Anal. (2013). DOI: 10.1093/imanum/drt037
7. Nepomnyaschikh, S.V.: Schwarz alternating method for solving the singular Neumann problem. Soviet J. Numer. Anal. Math. Modelling **5**(1), 69–78 (1990)
8. Nepomnyaschikh, S.V.: Decomposition and fictitious domains methods for elliptic boundary value problems. In: D.E. Keyes et al. (eds.) Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, pp. 62–72. SIAM, Philadelphia (1992)
9. Oswald, P.: Preconditioners for nonconforming discretizations. Math. Comput. **65**(215), 923–941 (1996)
10. Xu, J.: Iterative methods by space decomposition and subspace correction. SIAM Rev. **34**, 581–613 (1992)
11. Xu, J.: The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. Computing **56**(3), 215–235 (1996)

ASM-BDDC Preconditioners with variable polynomial degree for CG- and DG-SEM

C. Canuto[1], L. F. Pavarino[2], and A. B. Pieri[3]

1 Introduction

Discontinuous Galerkin (DG) methods for partial differential equations are well suited to treat nonconforming meshes and inhomogeneous polynomial orders required by hp-adaptivity. Their elementwise formulation permit us to consider complex meshes and the relaxation of the continuity constraints allows the polynomial order to be refined locally. However, DG discretizations lead to large and ill-conditioned algebraic systems. In this paper, we study a quasi-optimal preconditioner for the spectral element version of Discontinuous Galerkin methods. In particular, we focus on the interior penalty formulation of such DG schemes. For a review of the different classes of DG methods, the reader is referred to [2].

Recent endeavors in the domain decomposition community have lead to the development of additive [7] and multiplicative [1] Schwarz preconditioners for DG. Among additive Schwarz solvers, nonoverlapping methods such as BDDC (Balancing Domain Decomposition by Constraints) or FETI-DP (Dual-Primal Finite Element Tearing and Interconnecting) for DG have been designed [6] considering only variations on the subdomain size H or the element size h in a finite element context. Based on the pioneer work by [5, 9] and later [10], the BDDC algorithm was recently generalized to CG-SEM (continuous Galerkin spectral elements) in [12, 8]. Following the work in [11], and more recently [3], we make use of the ASM (*Auxiliary Space Method*) to derive a preconditioner for DG-SEM. The paper is organized as follows.

First, we generalize the BDDC preconditioners for CG-SEM studied in [12] to inhomogeneous polynomial distributions, where polynomial degrees is allowed to vary in different elements but we enforce the polynomial degree of the basis functions to match at the interface between elements.

Second, the ASM is presented and applied to derive a solver for DG-SEM based on the previous continuous solver. Once the Schur complement for the continuous problem is solved, the global continuous solution is readily obtained using exact local solvers. The discontinuous solution is then obtained solving the ASM problem. The resulting preconditioner is proved to have the same performances of the BDDC preconditioners for CG-SEM if the polynomial jumps are smooth enough.

[1] Politecnico di Torino, e-mail: ccanuto@calvino.polito.it .[2] Università di Milano e-mail: luca.pavarino@unimi.it .[3] Ecole Centrale de Lyon alexandre.pieri@ec-lyon.fr

In the last section, we present numerical simulations showing the robustness of the extended BDDC preconditioner with respect to polynomial jumps. The ASM-BDDC is finally tested by varying the number of spectral elements per subdomain H/h, the polynomial degree p and the viscosity coefficients.

The present work is an extension of [4].

2 Balancing Domain Decomposition by Constraints with inhomogeneous polynomial degrees

We consider the second-order elliptic problem with homogeneous Dirichlet boundary conditions

$$-\nabla\cdot(\mu\nabla u) = f \quad \text{in } \Omega\,, \qquad u = 0 \quad \text{on } \partial\Omega\,, \tag{1}$$

where $\Omega \subset \mathbb{R}^d$ ($d = 2,3$) is a bounded domain with Lipschitz boundary. Problem (1) admits a unique weak solution in $H_0^1(\Omega)$ if we assume that $f \in L^2(\Omega)$ and $\mu \in L^\infty(\Omega)$, with $\mu \geq \mu_0$ a.e. in Ω for a suitable constant $\mu_0 > 0$.

2.1 CG-SEM discretization for elliptic problems

Given a partition of $\overline{\Omega} = \bigcup_{k=1}^N \Omega_k$ into spectral elements, we define the continuous Galerkin space $\mathcal{V}_\delta^C = \{v : \Omega \to \mathbb{R} \mid \forall k,\ v_{|\Omega_k} \in \mathbb{P}_{p_k}(\Omega_k),\ v \in C^0(\Omega)\}$, that is the space of continuous elementwise polynomial functions. Problem (1) in its weak form is then:
Find $u \in H_0^1(\Omega)$ such that

$$a_c(u,v) = L(v) \quad \forall v \in \mathcal{V}_\delta^C, \tag{2}$$

where

$$a_c(u,v) = \sum_k \int_{\Omega_k} \mu(x)\nabla u\cdot\nabla v dx, \quad L(v) = \sum_k \int_{\Omega_k} f v dx.$$

Considering elliptic coefficients μ that are constant on each spectral element *i.e.* $\mu_{|\Omega_k} = \mu_k$, the bilinear form of problem (2) can be written

$$a_c(u,v) = \sum_k \mu_k a_c^k(u,v). \tag{3}$$

2.2 CG-SEM with locally varying polynomial degrees

The definition of $\mathscr{V}_\delta^C$ allows the polynomial degree to vary inside an element. However, the continuity constraint forces the polynomial degrees to match at the interface between two spectral elements, in the direction parallel to the interface. Therefore, the polynomial degree at the interface is enforced by the spectral element carrying the lowest polynomial degree. For a given polynomial order $\mathbf{p}$ on Ω_k, we introduce the nodal basis functions $\{\psi_{i_n}\}_{i_n=0\cdots p_n}$ formed by the (p_n+1) Lagrange interpolants at the Gauss-Legendre-Lobatto (GLL) nodes $\{x_{i_n}\}_{i_n=0\cdots p_n}$ in the n-th dimension. Considering a node $\mathbf{x}\in\Omega_k$, the following two configurations can occur:

- $\mathbf{x}\in\Omega_k/\partial\Omega_k$. In this case the basis function $\phi_{\mathbf{j}}$ relative to $\mathbf{x}$ is obtained by tensorial product of one-dimensional basis functions and $\phi_{\mathbf{j}}(\mathbf{x})=\Pi_{n=1}^d\ \psi_{j_n}(x_n)$.
- $\mathbf{x}\in\partial\Omega_k$. In this case, $\mathbf{x}$ lies on a face $F=\Omega_k\cap\Omega_{k'}$ normal to, lets say, the q-th dimension. The basis function $\phi_{\mathbf{j}}$ relative to $\mathbf{x}$ is built as $\phi_{\mathbf{j}}(\mathbf{x})=\psi_{j_q}(x_q)\Pi_{n\neq q}\psi_{j_n}^\perp(x_n^\perp)$. The functions $\left\{\psi_{j_n}^\perp\right\}$ — defined as the Lagrange interpolants at the GLL nods $\left\{x_n^\perp\right\}$ — are obtained by linear combinations of the $\left\{\psi_{j_n}\right\}$

$$\psi_{j_n}^\perp(x)=\sum_{i_m}\psi_{j_n}^\perp(x_m)\psi_{i_m}(x)=\sum_{i_m}\mathscr{C}_{nm}^k\psi_{i_m}(x).$$

  The nodes $\left\{x_n^\perp\right\}$ are given by the lowest GLL quadrature on the face F: $p_F=\min(p_k,p_{k'})$.

Problem (2) is now brought into the algebraic form

$$A\mathbf{u}=f, \tag{4}$$

where $A=\sum_{n=1}^N\mathscr{P}_n^tA^n\mathscr{P}_n$ and $\{A^n\}$ are the matrices representing the bilinear forms $a_c^n(.,.)$ of problem (3). The $\{\mathscr{P}_n\}$ are defined in terms of the coefficient $\{\mathscr{C}_{ij}^n\}$

$$\mathscr{P}_n=\begin{bmatrix}I & 0\\ 0 & \mathscr{C}^n\end{bmatrix},$$

provided that internal unknowns are all ordered before those of the interface. In the next section, we present the continuous solver relative to this algebraic system.

2.3 BDDC as a preconditioner for the Schur complement

In this section, we assume that the domain Ω is decomposed into nonoverlapping subdomains $\Omega=\bigcup_k\Omega^{(k)}$. Each subdomain $\Omega^{(k)}$ has diameter H_k and is composed of several spectral elements $\Omega^{(k)}=\bigcup_{m=1}^{N_k}\Omega^m$ having diameter h_k — we assume without loss of generality that the partition is spatially uniform inside a subdomain — so that H_k/h_k quantifies the number of spectral elements along a subdomain edge.

By partitioning the local degrees of freedom into interior (I) and interface (Γ) sets, and by further partitioning the latter into dual (Δ) and primal (Π) degrees of freedom, then the matrix $A^{(n)}$ relative to the restriction of $a_c(.,.)$ to the n-th subdomain $\Omega^{(n)}$ can be written as

$$A^{(n)} = \begin{bmatrix} A_{II}^{(n)} & A_{\Gamma I}^{(n)^T} \\ A_{\Gamma I}^{(n)} & A_{\Gamma\Gamma}^{(n)} \end{bmatrix} = \begin{bmatrix} A_{II}^{(n)} & A_{\Delta I}^{(n)^T} & A_{\Pi I}^{(n)^T} \\ A_{\Delta I}^{(n)} & A_{\Delta\Delta}^{(n)} & A_{\Pi\Delta}^{(n)^T} \\ A_{\Pi I}^{(n)} & A_{\Pi\Delta}^{(n)} & A_{\Pi\Pi}^{(n)} \end{bmatrix}. \tag{5}$$

The choice of primal and dual variables is discussed in [12]. In two dimensions, the primal variables reduce to the vertices of the subdomains while the dual ones correspond to the unknowns lying on an interface between two subdomains. Using the scaled restriction matrices defined in [12] and keeping the same notations, the BDDC preconditioner for the Schur complement of system (4) can be written as

$$M^{-1} = \widetilde{R}_{D,\Gamma}^T \widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma}, \tag{6}$$

where

$$\widetilde{S}_\Gamma^{-1} = R_{\Gamma\Delta}^T \left(\sum_{n=1}^N \begin{bmatrix} 0 & R_\Delta^{(n)^T} \end{bmatrix} \begin{bmatrix} A_{II}^{(n)} & A_{\Delta I}^{(n)^T} \\ A_{\Delta I}^{(n)} & A_{\Delta\Delta}^{(n)} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ R_\Delta^{(n)} \end{bmatrix} \right) R_{\Gamma\Delta} + \Phi S_{\Pi\Pi}^{-1} \Phi^T, \tag{7}$$

with the coarse matrix

$$S_{\Pi\Pi} = \sum_{n=1}^N R_\Pi^{(n)^T} \left(A_{\Pi\Pi}^{(n)} - \begin{bmatrix} A_{\Pi I}^{(n)} & A_{\Pi\Delta}^{(n)} \end{bmatrix} \begin{bmatrix} A_{II}^{(n)} & A_{\Delta I}^{(n)^T} \\ A_{\Delta I}^{(n)} & A_{\Delta\Delta}^{(n)} \end{bmatrix}^{-1} \begin{bmatrix} A_{\Pi I}^{(n)^T} \\ A_{\Pi\Delta}^{(n)^T} \end{bmatrix} \right) R_\Pi^{(n)}$$

and a matrix Φ mapping interface variables to primal degrees of freedom, given by

$$\Phi = R_{\Gamma\Pi}^T - R_{\Gamma\Delta}^T \sum_{n=1}^N \begin{bmatrix} 0 & R_\Delta^{(n)^T} \end{bmatrix} \begin{bmatrix} A_{II}^{(n)} & A_{\Delta I}^{(n)^T} \\ A_{\Delta I}^{(n)} & A_{\Delta\Delta}^{(n)} \end{bmatrix}^{-1} \begin{bmatrix} A_{\Pi I}^{(n)^T} \\ A_{\Pi\Delta}^{(n)^T} \end{bmatrix} R_\Pi^{(n)}.$$

Equation (7) means that we solve on each subdomain a problem with Neumann data for the dual variables and a coarse problem with matrix $S_{\Pi\Pi}$ for the primal variables.

Theorem 1. *The condition number κ_2 of the BDDC and FETI-DP preconditioned systems in 2D, using at least one primal vertex for each subdomain edge $F_\Omega \subseteq \Gamma$, satisfies the following bound:*

$$\kappa_2(M^{-1}\widehat{S}) \leq C \left(1 + \log\left(H \max_{F_K \subseteq \Gamma} \frac{p_{F_K}^2}{h_{F_K}} \right) \right)^2, \tag{8}$$

where p_{F_K} is the polynomial degree over an element edge F_K (we recall that if $F_K = \partial K \bigcap \partial K'$, then $p_{F_K} = \min(p_K, p_{K'})$ and the constant $C > 0$ is independent of p_{F_K}, h_{F_K}, H and the values of the coefficient μ of the elliptic operator.

This result (see [4] for a proof) states in particular that the preconditioned problem is scalable in the number of subdomains and robust with respect to jumps in the elliptic coefficients. Once we have a preconditioner for the Schur complement of the CG-SEM problem, we are able to build a global preconditioner for DG *via* the Auxiliary Space Method. This is the object of the next section.

3 Preconditioning DG with ASM-BDDC

3.1 DG-SEM discretization for elliptic problems

We recall that the weak form of problem (1) obtained choosing as Galerkin space $\mathscr{V}_\delta = \{v : \Omega \to \mathbb{R} \mid \forall k,\ v_{|\Omega_k} \in \mathbb{P}_{p_k}(\Omega_k),\ v \in L^2(\Omega)\}$, that is the space of discontinuous elementwise polynomial functions is given by:
Find $u \in H_0^1(\Omega)$ such that

$$a(u,v) = L(v) \quad \forall v \in \mathscr{V}_\delta, \tag{9}$$

where the bilinear form defined on $\mathscr{V}_\delta \times \mathscr{V}_\delta$ is

$$\begin{aligned} a_\delta(u,v) = &\sum_{K\in\mathscr{K}} \int_K \mu\, \nabla u \cdot \nabla v - \sum_{F\in\mathscr{F}} \mu_F \int_F \{\{\nabla u\}\}_F [\![v]\!]_F + \{\{\nabla v\}\}_F [\![u]\!]_F \\ &+ \sum_{F\in\mathscr{F}} \eta_F \mu_F \int_F [\![u]\!]_F [\![v]\!]_F \ , \end{aligned}$$

as well as the linear form $F(v) = \int_\Omega fv$ defined on $\mathscr{V}_\delta$. The jump $[\![.]\!]_F$ and average $\{\{.\}\}_F$ operators are the standard ones defined e.g. in [2] and the coefficients η_F and μ_F are defined as in [6]. Choosing an appropriate basis of $\mathscr{V}_\delta$, problem (9) is brought into its algebraic form and we are ready to apply the ASM preconditioning technique.

3.2 The auxiliary space method (ASM)

The Auxiliary Space Method (ASM) [11] gives a general framework for designing preconditioners of nonconforming discretizations, provided preconditioners for some related conforming discretizations are available. Hereafter, we recall the ASM formulation tailored to the current situation of interest, referring e.g. to [3] for the most general setting. We assume there exists a symmetric bilinear form $b_\delta(u,v)$ on $V_\delta \times V_\delta$ and a linear operator $Q_\delta^c : V_\delta \to V_\delta^c$ such that

$$a_\delta(v,v) \lesssim b_\delta(v,v) \quad \forall v \in V_\delta \tag{10}$$

and

$$b_\delta(v - Q_\delta^c v, v - Q_\delta^c v) \lesssim a_\delta(v,v) \qquad \forall v \in V_\delta \ . \tag{11}$$

Here and in the sequel, the symbol $\lesssim$ means $\leq c$ for a constant c bounded independently of δ in the admissible range of variability of δ. This implies the following algebraic results. Let $\mathbb{A}$ and B denote the matrices associated with the forms a_δ and b_δ once a basis in V_δ has been chosen; similarly, let A denote the matrix associated with the form $a = a_\delta$ restricted to V_δ^c, once a basis in V_δ^c has been chosen. Let Z be the matrix representing the inclusion $V_\delta^c \subset V_\delta$ in the chosen bases. In addition, assume that P_B^{-1} is a symmetric preconditioner for B and P_A^{-1} is a symmetric preconditioner for A, such that the following eigenvalue bounds hold:

$$\lambda_{\max}(P_B^{-1}B),\ \lambda_{\max}(P_A^{-1}A) \ \leq \ \Lambda_{\max}, \qquad \lambda_{\min}(P_B^{-1}B),\ \lambda_{\min}(P_A^{-1}A) \ \geq \ \Lambda_{\min} \ .$$

Then,

$$P_{\mathbb{A}}^{-1} := P_B^{-1} + Z P_A^{-1} Z^T \tag{12}$$

is a symmetric preconditioner for $\mathbb{A}$, such that

$$\kappa_2(P_{\mathbb{A}}^{-1}\mathbb{A}) \leq \frac{\Lambda_{\max}}{\Lambda_{\min}} \ . \tag{13}$$

Now, we choose for P_A^{-1} the global BDDC-based preconditioner defined according to [13]

$$P_A^{-1} = \begin{pmatrix} I & -A_{II}^{-1}A_{I\Gamma} \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{II}^{-1} & 0 \\ 0 & M^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -A_{\Gamma I}A_{II}^{-1} & I \end{pmatrix} \ , \tag{14}$$

where M^{-1} is the BDDC preconditioner of equation (6). The subscript Γ means that we consider the unknowns lying on the Schur skeleton while the subscript I is linked to internal unknowns (inside a subdomain). In the last section, we present some numerical results showing the robustness of preconditioners (6) and (12).

4 Numerical results and conclusion

We present two test cases that illustrate the robustness and quasi-optimality of both preconditioners P_A^{-1} and $P_{\mathbb{A}}^{-1}$. First, the number of spectral elements is fixed and we consider both jumping elliptic coefficients and polynomial degrees, see Figure 1. The results are presented in Table 1, where it is shown that the condition number $\kappa_2(P_A^{-1}A)$ is quite insensitive to moderate jumps in the polynomial degree such as $p \to p+2 \to p+4$. Discontinuities in the elliptic coefficients are managed quite well by the ASM-BDDC preconditioner for minor variations in the polynomial degree. We also study the sensitivity of κ_2 to simultaneous variations in h and p. In particular, setting $H = 1$ (that is the continuous solver is exact), Table 2 shows that

the condition number of the ASM-BDDC remains $O(1)$ in agreement with bound (8) in Theorem 1. Lastly, a case with rectangular spectral elements is investigated, see Figure 2. We consider a diadic evolution of spectral elements width h as 2^{-i} for $i = 1, \cdots, 5$ with a uniform polynomial degree. The results are presented in Figure 2 for both $k_2(P_{\mathbb{A}}^{-1}\mathbb{A})$ and $\kappa_2(P_A^{-1}A)$.

As a conclusion, this paper presents a new way of preconditioning DG-SEM systems based on an available preconditioner for CG-SEM. The ASM applied to such a global BDDC-based preconditioner provides a solver for DG that is still $0(H\log(\max\frac{p_K^2}{h_K}))$ but it also introduces a dependence on the maximal polynomial jump and elliptic coefficients. However, we show numerically that for moderate polynomial jumps, the preconditioner is scalable and quasi-optimal.

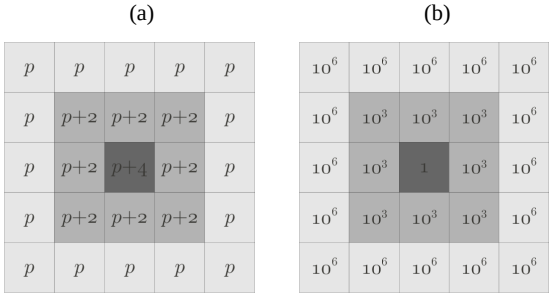


Fig. 1 Test case with 5×5 subdomains. Polynomial degrees and elliptic coefficients obey a pyramidal distribution, where p increases toward the center of the domain (a), the elliptic coefficient μ decreases (b).

Table 1 Condition numbers for increasing polynomial degree p with nonuniform (uniform in brackets) polynomial distribution and jumping elliptic coefficients given in Fig. 1, with 5×5 subdomains and $H/h = 1$.

Degree p	BDDC $\kappa_2(P_A^{-1}A)$	ASM-BDDC $\kappa_2(P_{\mathbb{A}}^{-1}\mathbb{A})$
2	2.34 (1.47)	5.95 (5.09)
4	3.37 (2.64)	6.31 (5.71)
6	4.20 (3.56)	6.54 (6.20)
8	4.89 (4.33)	6.70 (6.50)
10	5.49 (4.99)	6.83 (6.70)
12	6.02 (5.56)	6.94 (6.82)

References

1. Antonietti, P.F., Ayuso, B.: Schwarz domain decomposition preconditioners for Discontinuous Galerkin approximations of elliptic problems: non-overlapping case. ESAIM: Math. Model. Num. **41**(01), 21–54 (2007)
2. Arnold, D.N., Brezzi, F., Cockburn, B., Marini, L.D.: Unified analysis of Discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal. **39**(5), 1749–1779 (2002)

Table 2 Condition number of the preconditioned DG matrix for increasing polynomial degree p with uniform polynomial distribution and increasing h, so that the ratio p^2/h is maintained approximatively constant. Uniform elliptic coefficients $\mu_K = 1$. Results for one subdomain $H = 1$.

Degree p	# elements	$\frac{p^2}{h}$	ASM $\kappa_2(P_{\mathbb{A}}^{-1}\mathbb{A})$
2	25^2	100	5.10
3	10^2	90	5.44
4	6^2	96	5.82
5	4^2	100	6.07
6	3^2	108	6.25

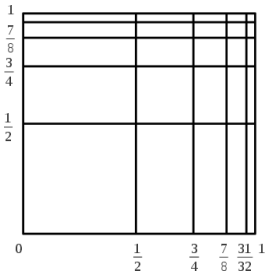


Fig. 2 Test case with uniform polynomial degree and diadic mesh in h. The ratio H/h is kept equal to 1, meaning one element per subdomain. Condition number of the preconditioned DG matrix for this configuration. Uniform elliptic coefficients $\mu_K = 1$. Results for one element per subdomain $H/h = 1$.

p	$\kappa_2(P_A^{-1}A)$	$\kappa_2(P_{\mathbb{A}}^{-1}\mathbb{A})$
2	4.12	14.13
4	4.09	13.46
6	4.75	12.89
8	5.43	12.50
10	6.07	12.24
12	6.65	12.08

3. Brix, K., Campos-Pinto, M., Canuto, C., Dahmen, W.: Multilevel preconditioning of Discontinuous-Galerkin spectral element methods. part i: Geometrically conforming meshes. IGPM Preprint 355, RWTH Aachen (2013). Submitted. arXiv:1301.6768
4. Canuto, C., Pavarino, L.F., Pieri, A.B.: BDDC preconditioners for continuous and discontinuous Galerkin methods using spectral/hp elements with variable polynomial degree. IMA J. Numer. Anal. (2013). DOI: 10.1093/imanum/drt037
5. Dohrmann, C.R.: A preconditioner for substructuring based on constrained energy minimization. SIAM J. Sci. Comput. **25**(1), 246–258 (2003)
6. Dryja, M., Galvis, J., Sarkis, M.: BDDC methods for Discontinuous Galerkin discretization of elliptic problems. J. Complexity **23**(4), 715–739 (2007)
7. Feng, X., Karakashian, O.A.: Two-level additive Schwarz methods for a Discontinuous Galerkin approximation of second order elliptic problems. SIAM J. Numer. Anal. **39**(4), 1343–1365 (2001)
8. Klawonn, A., Pavarino, L.F., Rheinbach, O.: Spectral element FETI-DP and BDDC preconditioners with multi-element subdomains. Comput. Method. Appl. M. **198**(3), 511–523 (2008)
9. Mandel, J., Dohrmann, C.R.: Convergence of a balancing domain decomposition by constraints and energy minimization. Numer. Linear Algebr. **10**(7), 639–659 (2003)
10. Mandel, J., Dohrmann, C.R., Tezaur, R.: An algebraic theory for primal and dual substructuring methods by constraints. Appl. Numer. Math. **54**(2), 167–193 (2005)
11. Oswald, P.: Preconditioners for nonconforming discretizations. Math. Comput. **65**(215), 923–942 (1996)
12. Pavarino, L.F.: BDDC and FETI-DP preconditioners for spectral element discretizations. Comput. Method. Appl. M. **196**(8), 1380–1388 (2007)
13. Toselli, A., Widlund, O.: Domain decomposition methods-algorithms and theory, vol. 34. Springer (2004)

Domain decomposition in shallow-water modelling for practical flow applications

Mart Borsboom[1], Menno Genseberger[1], Bas van 't Hof[2], and Edwin Spee[1]

1 Introduction

For the simulation of flows in rivers, lakes, and coastal areas for the executive arm of the Dutch Ministry of Infrastructure and the Environment the shallow-water solver SIMONA is being used [1]. Applications range from operational forecasting of flooding of the Dutch coast [3] and big lakes [7], to the assessment of primary water defences (coast, rivers, and lakes). These applications require a robust and efficient modelling framework with extensive modelling flexibility and good parallel performance.

About two decades ago, a parallel implementation of SIMONA was developed [10, 11] based on domain decomposition with maximum overlap. In the same period, non-overlapping domain decomposition with optimized coupling was considered for Delft3D-FLOW [2], a shallow-water solver that is numerically very similar to SIMONA. More recently, ideas of the latter were adapted for incorporation in SIMONA for enhanced modelling flexibility and parallel performance. This will be the subject of the present paper.

The paper is organized as follows. The numerical approach for modelling shallow-water flow as implemented in SIMONA is outlined in section 2. In section 3 we show how domain decomposition has been incorporated and which refinements have been made. The parallel performance of the modified method is illustrated in section 4 for two practical flow problems from civil engineering.

2 ADI-type shallow-water solvers

The shallow-water equations consist of a depth-integrated continuity equation and two horizontal momentum equations. Vertical momentum is replaced by the hydrostatic pressure assumption, i.e., the vertical variation of the pressure is assumed to depend solely on hydrostatic forces as determined by the position of the free surface. For the numerical solution of the shallow-water equations SIMONA applies a so-called alternating direction implicit (ADI) method to integrate the equations numerically in time, using an orthogonal staggered grid with horizontal curvilinear coordinates ξ and η [1].

[1] Deltares, Delft, The Netherlands, e-mail: {Mart.Borsboom}{Menno.Genseberger}{Edwin.Spee}@deltares.nl .[2] VORtech Computing, Delft, The Netherlands, e-mail: bas.vanthof@vortech.nl

In the ADI method, each time step is split in two stages of half a time step. In the first stage, the water-level gradient is taken implicitly in the ξ-momentum equation and explicitly in the η-momentum equation. The mass fluxes in the continuity equation are taken implicitly/explicitly in ξ- and η-direction as well, allowing the implicit terms to be combined to uncoupled tridiagonal systems of equations in ξ-direction for the water level at the intermediate time level. In contrast, the evaluation of the horizontal convection terms and viscosity terms are respectively explicit and implicit in the ξ- and η-momentum equation. In the second stage of the time step, the implicit and explicit discretisations are interchanged. For stability, derivatives in vertical direction and the bottom friction term are always integrated implicitly.

The ADI method requires the use of fairly small time steps to avoid excessive splitting errors:

$$\frac{u\Delta t}{\Delta x_\xi} \leq O(1)\ ,\ \frac{v\Delta t}{\Delta x_\eta} \leq O(1)\ ,\ \frac{\sqrt{gh}\Delta t}{\Delta x_\xi} \leq O(10)\ ,\ \text{and}\ \frac{\sqrt{gh}\Delta t}{\Delta x_\eta} \leq O(10)\ . \qquad (1)$$

Here, $\Delta x_\xi, \Delta x_\eta$ are the grid sizes and u, v the velocities in the two horizontal curvilinear coordinate directions ξ and η, Δt is the time step, h the local water depth, and g the acceleration due to gravity ($\sqrt{gh}$ is the shallow-water wave celerity). Because of the conditions (1), the discretized equations to be solved have a fairly high diagonal dominance horizontally. This enables the use of semi-explicit iterative methods horizontally, such as red-black Jacobi to solve implicit convection and viscosity. For the same reason, horizontal domain decomposition with explicit coupling, if designed properly, can be very efficient. We remark that in the vertical direction grid sizes $\ll \Delta x_\xi, \Delta x_\eta$ are used and the systems of equations are much stiffer. Vertical derivatives are therefore always integrated implicitly in time.

3 Domain decomposition techniques for ADI-type shallow-water solvers

About two decades ago, a parallel implementation of SIMONA was developed [10, 11] using a multi-domain version of the ADI method with Dirichlet-Dirichlet coupling and maximum overlap to ensure fast convergence. This approach is still applied in the 2006 version of SIMONA. Later on, for modelling flexibility, the possibility to use different grid resolutions per subdomain has been introduced. For such a situation it is not that easy to deal with an overlap between subdomains. Therefore, the overlap was removed. This concerns the overlap of the physical area of the subdomains, i.e., the area containing the inner grid cells. For the implementation of boundary conditions and coupling conditions, virtual grid cells were added outside the physical areas along boundaries and DD interfaces. So although the subdomains do not overlap, the subdomain grids do. Unfortunately, a Dirichlet-Dirichlet coupling with minimal overlap (only the virtual grid cells overlap) has a very slow rate of convergence. See also panel (b) of Fig. 1. By re-using ideas from a non-overlapping domain decomposition approach with optimized coupling

for Delft3D-FLOW [2], the good convergence behavior has been restored. This approach is implemented since 2010 in SIMONA.

To illustrate how convergence errors due to domain decomposition propagate from one subdomain to another in a multi-domain ADI-type shallow-water solver, we consider a uniform grid of size Δx_ξ, a uniform depth h, and assume a small surface elevation ζ and flow velocity u. The implicit systems in the ξ-direction at the first half time step from t^n to $t^{n+1/2}$ are then of the form (discretized continuity equation and momentum equation):

$$\frac{\zeta_i^{n+1/2}-\zeta_i^n}{\Delta t/2}+h\frac{u_{i+1/2}^{n+1/2}-u_{i-1/2}^{n+1/2}}{\Delta x_\xi}=\dots\,,\quad \frac{u_{i+1/2}^{n+1/2}-u_{i+1/2}^n}{\Delta t/2}+g\frac{\zeta_{i+1}^{n+1/2}-\zeta_i^{n+1/2}}{\Delta x_\xi}=\dots\,. \tag{2}$$

At the second half time step from $t^{n+1/2}$ to t^{n+1}, equations in η-direction (j-index) are obtained. By eliminating $u_{i+1/2}^{n+1/2}$, the two equations (2) can be combined to:

$$\zeta_i^{n+1/2}-CFL^2\left(\zeta_{i+1}^{n+1/2}-2\,\zeta_i^{n+1/2}+\zeta_{i-1}^{n+1/2}\right)=\dots\,, \tag{3}$$

with CFL number $CFL=\sqrt{gh}\,\Delta t/(2\Delta x_\xi)$.

To study the behavior of (3) in a DD framework, we consider the homogeneous equation that is satisfied by the DD convergence error $\delta\zeta_i^{n+1/2,m}=\zeta_i^{n+1/2,m}-\zeta_i^{n+1/2}$, with $\zeta_i^{n+1/2}$ the solution that is sought and $\zeta_i^{n+1/2,m}$ its iteratively determined approximation at iteration m:

$$\delta\zeta_i^{n+1/2,m}-CFL^2\left(\delta\zeta_{i+1}^{n+1/2,m}-2\,\delta\zeta_i^{n+1/2,m}+\delta\zeta_{i-1}^{n+1/2,m}\right)=0\,. \tag{4}$$

The inhomogeneous perturbation of $\delta\zeta_i^{n+1/2,m}$ comes from the boundaries of the subdomains where information is updated explicitly (Schwarz algorithm). Equation (4) determines how that information spreads across a subdomain and reaches the opposite subdomain boundary. This becomes clear from the solution of (4), which is of the form:

$$\delta\zeta_i^{n+1/2,m}=C_{LR}\lambda^i+C_{RL}\lambda^{-i}\,, \tag{5}$$

with $\lambda=(CFL^2+1/2-\sqrt{CFL^2+1/4})/CFL^2$. The solution consists of the superposition of two modes: one decaying from left to right and one decaying from right to left. Panel (a) in Fig. 1 illustrates this for a subdomain of 8 grid cells at $CFL=2$ (green), $CFL=5$ (red), and $CFL=10$ (blue). For $CFL\ll 1$, we have $\lambda\approx 1/CFL^2$. At such a high decay rate per grid cell, which is due to the large diagonal dominance of (4), a Dirichlet-Dirichlet coupling is efficient. For $CFL\gg 1$, however, we have $\lambda\approx 1-CFL^{-1}$ and hence a much lower decay rate. A Dirichlet-Dirichlet coupling is then not efficient anymore, unless a large overlap is used to compensate for the low decay rate. This is illustrated in panel (b) and (c) of Fig. 1.

A much larger DD convergence speed is obtained by only transfering from left to right (right to left) the information that is moving in that direction. This is realized by the coupling:

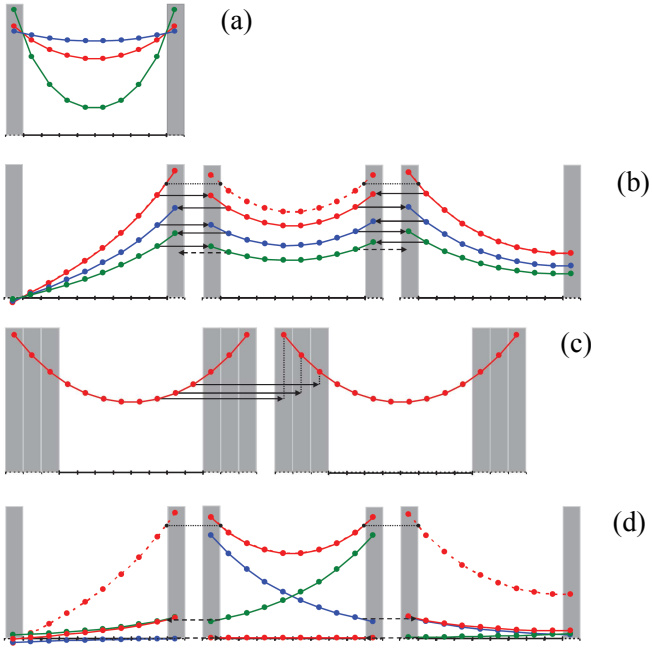


Fig. 1 Behavior of convergence error $\delta\zeta_i^{n+1/2,m}$ in subdomains consisting of 8 inner grid cells (white) and 1, 2, or 3 added virtual grid cells (grey) that overlap with inner grid cells of neighbouring subdomains: (a) inside a subdomain at $CFL = 2$ (green), $CFL = 5$ (red), and $CFL = 10$ (blue); (b) across 3 subdomains at $CFL = 5$ with Dirichlet boundary condition left, Neumann boundary condition right, and multiplicative Schwarz Dirichlet-Dirichlet coupling with minimal overlap in between (red, blue, green indicate subsequent DD iterations); (c) enhancement of DD convergence with Dirichlet-Dirichlet coupling when using a larger overlap (increasingly longer dotted lines indicate error reduction for 1-, 2-, and 3-cell overlap); (d) across 3 subdomains with optimized multiplicative Schwarz based on the decomposition of the convergence error (red lines) in its two solution modes (blue and green lines), cf. (5). Note that in (b, c) the arrows indicate the transfer of Dirichlet values from an inner grid cell to a virtual grid cell; in (d) the arrows indicate the transfer of optimized coupling information from interface to interface.

$$\begin{aligned}&(CFL+1/2)\delta\zeta_{i_R}^{n+1/2,m+1} - (CFL-1/2)\delta\zeta_{i_R+1}^{n+1/2,m+1}\\ &= (CFL+1/2)\delta\zeta_{i_L-1}^{n+1/2,m} - (CFL-1/2)\delta\zeta_{i_L}^{n+1/2,m}\,,\end{aligned} \tag{6}$$

with i_R the index of the left virtual grid cell of the subdomain right of the DD interface under consideration, and with i_L the index of the right virtual grid cell of the subdomain left. Notice the explicit nature of the coupling: the solution of domain L at previous iteration m determines the value (right-hand side of (6)) of the condition to be imposed at the left boundary of domain R during next iteration $m+1$ (left-hand side of (6)). An equivalent procedure is used for the transfer of coupling information in the other direction, from domain R to domain L.

Panel (d) of Fig. 1 illustrates the high DD convergence rate that can be obtained with an optimized coupling; the convergence speed is about as high as would be obtained with a Dirichlet-Dirichlet coupling with maximum overlap (of half a subdomain, cf. panel (c)). However, because of the overlap, the amount of work per iteration in the latter would be twice as large. Furthermore, as mentioned before, it can not be combined easily with local grid refinements for which the grid cells in the overlap do not coincide, contrary to the situation in panel (c).

The fast DD convergence speed that for diagonally dominant problems can be obtained with an optimized explicit local DD coupling (optimized Schwarz), and the link with absorbing boundary conditions, is well known [8, 5, 4, 9, 6]. Because the splitting applied in the ADI method leads to independent 1D problems, we have the advantage that the optimized coupling can not only easily be determined for constant Δx_ξ and h, as we did here, but also for the general case, by means of the LU decomposition of the resulting tridiagonal systems that are of the form (3), but with space- and time-varying coefficients. The bidiagonal L-matrices describe the decay of the solution in increasing i- (or j-) direction. Their last rows determine the combinations of pairs of ζ's at the subdomain interface (one ζ in a virtual grid cell, the other ζ in the adjacent inner grid cell) that do not specify this part of the solution, and hence only specify solution modes decaying in decreasing i- (or j-) direction. Transfering these combinations in decreasing i- (or j-) direction across DD interfaces (the variable-coefficient generalization of (6)) therefore ensures maximum DD convergence speed. Likewise for the bidiagonal U-matrices and the exchange of coupling information in the other direction.

4 Applications

There are many application areas of SIMONA. Here we present two examples. First we show the effect of the optimized coupling without overlap for a schematic model of the river Waal in the Netherlands. This schematic model has a simple geometric shape such that load balancing is straightforward. Second we show the parallel performance of the approach for DSCM, a huge real-life hydrodynamic model in which both load balancing and number of unknowns are an issue.

For the experiments we considered the following hardware:

- H4 linux-cluster at Deltares, nodes interconnected with Gigabit Ethernet, each node contains 1 AMD dual-core Athlon X2 5200B processor with 2.7 GHz per core,
- H4+ linux-cluster at Deltares, nodes interconnected with Gigabit Ethernet, each node contains 1 Intel quad-core i7-2600 processor with 3.4 GHz per core and hyperthreading (so effectively 8 threads are used on 4 cores), and
- Lisa linux-cluster at SURFsara, nodes interconnected with Infiniband, each node contains 2 Intel quad-core Xeon L5520 processors with 2.3 GHz per core.

On the H4 linux-cluster both the 2006 and 2010 version of SIMONA were used. On the H4+ and Lisa linux-cluster the 2010 version of SIMONA was used. Recall

(see section 3) that the 2006 version uses Dirichlet-Dirichlet coupling and maximum overlap where the 2010 version uses optimized coupling without overlap.

4.1 Schematic model of river Waal

To study the effect of lowering the groynes on design flood level, in [12] a schematised river reach was used that was based on characteristic dimensions of river Waal in the Netherlands. Here, for the performance tests we will use the detailed model of [12] in which the groynes are represented as bed topography (see Fig. 2).

The detailed model is a symmetrical compound channel of 30 km length including floodplain (width of 1200 m) and main channel (width of 600 m). We apply a depth averaged version of SIMONA. The floodplain is schematised with grid cells of 2 m x 4 m and the main channel with grid cells of 2 m x 2 m, resulting in more than 9 million unknowns. A time step of 0.015 minutes is used, resulting in 12000 time steps for the 3 hour simulation that we consider here for the performance tests.

From Fig. 2 it can be observed that, in general, SIMONA scales well. Furthermore, on the H4 linux-cluster the 2010 version of SIMONA is about 20-30 % faster than the 2006 version. This additional work can be explained from the overlap in the 2006 version which is not in the 2010 version (see section 3). The difference in performance for the 2010 version of SIMONA on H4, Lisa, and H4+ linux-cluster is because of the different hardware.

4.2 Next generation Dutch Continental Shelf Model (DCSM)

The current generation of nested SIMONA models used for predicting water levels along the Dutch coast in an operational mode (see [3]) already require high performance computing. At the Lisa linux-cluster parallel performance of the 2010 version of SIMONA was tested for a next generation version of the DCSM (North Sea and adjacent region of the North Atlantic). This 3D (10 layer) higher resolution model includes salinity and temperature stratification processes which are essential for simulating among others the spread of the freshwater Rhine plume along the Dutch coast. This new model requires a huge computational effort but simulation times cannot increase for operational purposes. Although the North Sea model has an irregular geometry which is not ideal for scalability, performance tests at Lisa showed linear scalability up to 100 processors. The left panel of Fig. 3 shows the partitioning of the domain in 96 subdomains of (about) the same number of grid cells that is obtained by applying orthogonal recursive bisection (ORB). The right panel shows the parallel performance on the Lisa linux-cluster as a function of the number of subdomains and cores, for partitionings in strips and by means of ORB. The results show an optimal speed-up for the ORB partitioning and a small decay in performance for the larger strip decompositions. The latter is due to the shape of the strips. The strips become very thin with widths of less than a dozen grid cells as the number of domains increases, which affects the validity of the applied local coupling optimization.

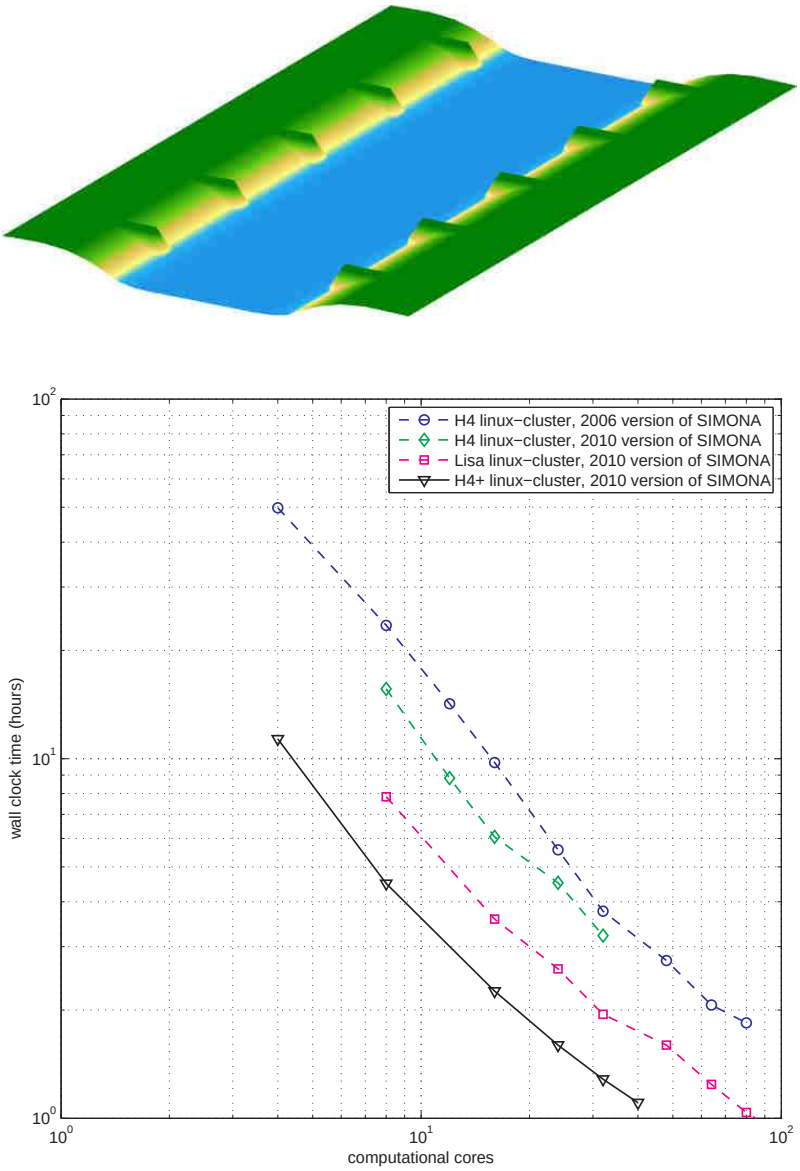


Fig. 2 Schematic model of river Waal: an excerpt of the model including part of the floodplain (top), parallel performance for different versions of SIMONA and on different hardware (bottom).

Acknowledgements We thank SURFsara (www.surfsara.nl) for their support in using the Lisa linux-cluster.

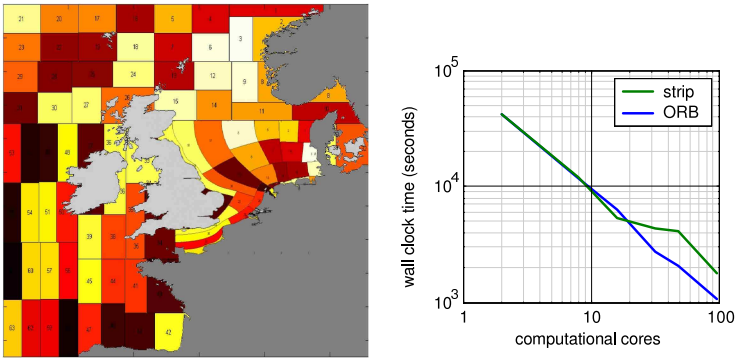


Fig. 3 DCSM. Left: partitioning of computational domain in 96 subdomains using the orthogonal recursive bisection (ORB) method. Right: parallel performance on Lisa linux-cluster for partitionings in vertical strips and ORB partitionings.

References

1. SIMONA WAQUA/TRIWAQ - two- and three-dimensional shallow-water flow model. (2012) URL {http://apps.helpdeskwater.nl/downloads/extra/simona/release/doc/techdoc/waquapublic/sim1999-01.pdf}
2. De Goede, E.D., Groeneweg, J., Tan, K.H., Borsboom, M.J.A., Stelling, G.S.: A domain decomposition method for the three-dimensional shallow water equations. Simulation Practice and Theory **3**, 307–325 (1995)
3. De Kleermaeker, S.H., Verlaan, M., Kroos, J., Zijl, F.: A new coastal flood forecasting system for the Netherlands. In: T. Van Dijk (ed.) Hydro12 Conference, Rotterdam, The Netherlands, 2012. Hydrographic Society Benelux (2012). URL {http://proceedings.utwente.nl/246}
4. Dolean, V., Lanteri, S., Nataf, F.: Convergence analysis of additive Schwarz for the Euler equations. Appl. Numer. Math. **49**(2), 153–186 (2004)
5. Engquist, B., Zhao, H.K.: Absorbing boundary conditions for domain decomposition. Appl. Numer. Math. **27**(4), 341–365 (1998)
6. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**(2), 699–731 (2006)
7. Genseberger, M., Smale, A., Hartholt, H.: Real-time forecasting of flood levels, wind driven waves, wave runup, and overtopping at dikes around Dutch lakes. In: F. Klijn, T. Schweckendiek (eds.) 2nd European Conference on FLOODrisk Management, Rotterdam, The Netherlands, 2012, Comprehensive Flood Risk Management, pp. 1519–1525. Taylor & Francis Group (2013)
8. Japhet, C., Nataf, F., Roux, F.X.: Extension of a coarse grid preconditioner to non-symmetric problems. In: J. Mandel, C. Farhat, X.C. Cai (eds.) Domain Decomposition Methods 10, *Contemporary Mathematics*, vol. 218, pp. 279–286. AMS (1998)
9. Maday, Y., Magoulès, F.: Absorbing interface conditions for domain decomposition methods: A general presentation. Comput. Meth. Appl. Mech. Eng. **195**(29–32), 3880–3900 (2006)
10. Roest, M.R.T.: Partitioning for parallel finite difference computations in coastal water simulation. Ph.D. thesis, Delft University of Technology, The Netherlands (1997)
11. Vollebregt, E.A.H.: Parallel software development techniques for shallow water models. Ph.D. thesis, Delft University of Technology, The Netherlands (1997)
12. Yossef, M.F.M., Zagonjolli, M.: Modelling the hydraulic effect of lowering the groynes on design flood level. Tech. Rep. 1002524-000, Deltares (2010)

Space-Time Domain Decomposition with Finite Volumes for Porous Media Applications

Paul-Marie Berthe[1], Caroline Japhet[2], and Pascal Omnes[1]

1 Introduction

In the context of simulating flow and transport in porous media (e.g. for the assessment of nuclear waste repository safety), two main challenges must be taken into account : the heterogeneity of the medium with physical properties ranging over several orders of magnitude, and widely differing space-time scales. Solving these features accurately requires very fine meshes or well-adapted and highly nonconforming meshes. On the one hand, one possible approach is to use non-overlapping domain decomposition which leads to efficient parallel algorithms with local adaptation in both space and time. The Optimized Schwarz Waveform Relaxation method (OSWR) [3, 2] with the Discontinuous Galerkin (DG) scheme in time [4] is a solution procedure which allows local time stepping. On the other hand, the finite volume schemes of DDFV type (Discrete Duality Finite Volumes) for diffusion problems [5] allow highly nonconforming meshes. Finally, [6] presents a strategy which is well adapted to domain decomposition for coupling upwind discretization of the convection with diffusion in the context of a finite volume method. In this paper, we extend the OSWR method to the DDFV scheme for advection-diffusion problems, using the strategy of [6]. The method is proven to be well posed and we prove the convergence of the iterative algorithm.

We consider the following transport equation in a porous medium :

$$\begin{aligned} \mathscr{L}c = \omega\partial_t c - \nabla\cdot(\boldsymbol{K}\nabla c - \boldsymbol{b}c) = f, \quad \text{in } \Omega\times(0,T), \\ c(.,0) = c_0, \ \text{in } \Omega, \end{aligned} \tag{1}$$

where Ω is an open bounded polygonal subset of $\mathbb{R}^2$, c is the concentration (e.g. of radionuclides) and f the source term. Equation (1) is supplemented with homogeneous Dirichlet boundary conditions. We assume that Ω is decomposed into non-overlapping subdomains. For the sake of simplicity, we present the method in the case of two polygonal subdomains Ω_L and Ω_R with interface $\Gamma := \partial\Omega_L \cap \partial\Omega_R$ (the method can be extended to the many subdomain case). We assume that the possible discontinuities of the porosity coefficient ω, the tangential component of the advection velocity $\boldsymbol{b}$ and the anisotropic diffusion matrix $\boldsymbol{K}$ are along Γ. In the sequel, the subscripts and superscripts L (resp. R) refer to Ω_L (resp. Ω_R).

[1] CEA, DEN, DM2S-STMF, F-91191 Gif-sur-Yvette Cedex, France. Université Paris 13, LAGA, F-93430, Villetaneuse, France. e-mail: {berthe}{omnes}@math.univ-paris13.fr ·
[2] Université Paris 13, LAGA, UMR 7539, F-93430, Villetaneuse, France. INRIA Paris-Rocquencourt, BP 105, 78153 Le Chesnay, France, e-mail: Caroline.Japhet@inria.fr

The initial problem (1) is equivalent to a system of subproblems defined on Ω_L and Ω_R with the following physical transmission conditions on Γ: $[c]_\Gamma = 0$ and $[(\boldsymbol{K}\nabla c - \boldsymbol{b}c)\cdot\boldsymbol{n}]_\Gamma = 0$, where $[v]_\Gamma$ denotes the jump of v through Γ and $\boldsymbol{n}$ a normal vector to Γ. These interface conditions can also be written, through Robin interface operators $\mathscr{B}_L$ and $\mathscr{B}_R$, under the equivalent form

$$[\mathscr{B}_L c]_\Gamma = [\mathscr{B}_R c]_\Gamma = 0, \tag{2}$$

$$\text{with} \quad \mathscr{B}_L = (\boldsymbol{K}\nabla c - \boldsymbol{b}c)\cdot\boldsymbol{n}_L + \lambda_L \quad , \quad \mathscr{B}_R = (\boldsymbol{K}\nabla c - \boldsymbol{b}c)\cdot\boldsymbol{n}_R + \lambda_R, \tag{3}$$

where $\boldsymbol{n}_L$ (resp. $\boldsymbol{n}_R$) is the outward normal to Ω_L (resp. Ω_R) and λ_L (resp. λ_R) a strictly positive function in $L^\infty(\Gamma)$.

Then, an OSWR algorithm [3, 2] for solving problem (1) is:

$$\begin{cases} \mathscr{L} c_L^{(\ell+1)} = f & \text{in } \Omega_L \times (0,T) \\ c_L^{(\ell+1)}(.,0) = c_0 & \text{in } \Omega_L \\ \mathscr{B}_L c_L^{(\ell+1)} = \mathscr{B}_L c_R^{(\ell)} & \text{on } \Gamma \times (0,T) \end{cases} \quad \begin{cases} \mathscr{L} c_R^{(\ell+1)} = f & \text{in } \Omega_R \times (0,T) \\ c_R^{(\ell+1)}(.,0) = c_0 & \text{in } \Omega_R \\ \mathscr{B}_R c_R^{(\ell+1)} = \mathscr{B}_R c_L^{(\ell)} & \text{on } \Gamma \times (0,T) \end{cases} \tag{4}$$

where λ_L and λ_R optimize the convergence factor of (4), see [2, 8, 9].

In Section 2, we present the DDFV scheme for the advection–diffusion problem in the global domain Ω. Then, in Section 3, we describe the multidomain DDFV scheme. Section 4 is devoted to the OSWR algorithm for the DDFV scheme. Finally in Section 5, we present numerical results.

2 The DDFV scheme for advection-diffusion problems

In this part, we present the DDFV scheme for Problem (1). This scheme uses unknowns at the centers of the cells of a primal mesh and at their vertices. These vertices are considered as the centers of dual cells, obtained by joining the centers of the surrounding primal cells through the edge midpoints. This construction is sufficiently general to be able to treat non-conforming meshes, see Fig. 1 (left) where the primal (resp. dual) nodes are in black (resp. red), and T_{i_1} (resp. P_{k_1}) is an example of primal (resp. dual) cell. Using these supplementary vertex unknowns is the price to pay to be able to use arbitrary meshes [5]. We split $(0,T)$ into time intervals $I_n := (t_{n-1}, t_n)$ and define $\Delta t_n := t_n - t_{n-1}$. We denote by $c_{i_1}^n$ (resp. $c_{k_1}^n$) an approximation of c at time t_n in the cell T_{i_1} (resp. P_{k_1}). Restricting the presentation to the lowest order DG scheme in time, equation (1) can be discretized on each time interval and on each primal cell T_{i_1} by

$$\omega_{i_1}\frac{c_{i_1}^n - c_{i_1}^{n-1}}{\Delta t_n} - \frac{1}{|T_{i_1}|}\sum_{A_j \subset \partial T_{i_1}} |A_j| F_{i_1 j}^n = f_{i_1}^n := \frac{1}{\Delta t_n |T_{i_1}|}\int_{I_n}\int_{T_{i_1}} f(\boldsymbol{x},t)\,d\boldsymbol{x}\,dt, \tag{5}$$

and on each inner dual cell P_{k_1} by

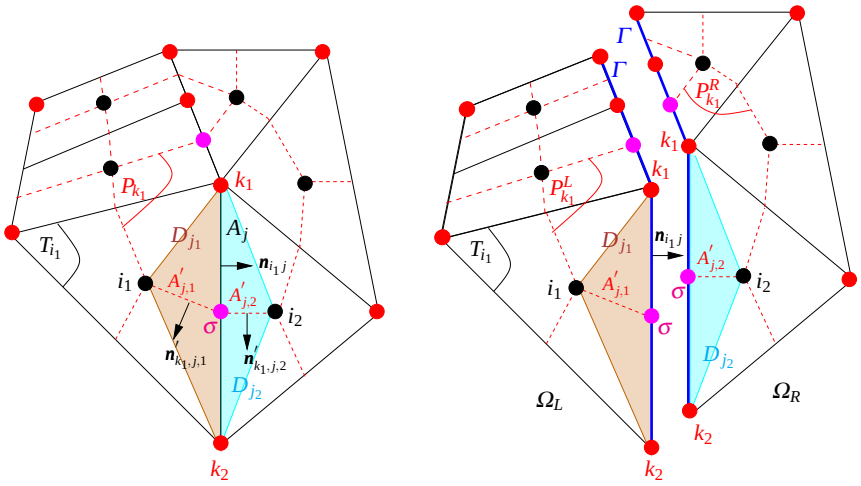


Fig. 1 DDFV primal (solid lines), dual (dashed lines) and half-diamond cells (filled triangles): interior (left) and interface (right) cells.

$$\omega_{k_1}\frac{c^n_{k_1}-c^{n-1}_{k_1}}{\Delta t_n}-\frac{1}{|P_{k_1}|}\sum_{A'_{j,\alpha}\subset\partial P_{k_1}}|A'_{j,\alpha}|F^n_{k_1 j,\alpha}=f^n_{k_1}:=\frac{1}{\Delta t_n|P_{k_1}|}\int_{I_n}\int_{P_{k_1}}f(\boldsymbol{x},t)d\boldsymbol{x}dt. \quad (6)$$

In (6), the subscript $\alpha\in\{1,2\}$ refers to the local numbering i_1,i_2, and ω_{k_1} is defined by

$$|P_{k_1}|\omega_{k_1}=|P_{k_1}\cap\Omega_L|\omega^L_{k_1}+|P_{k_1}\cap\Omega_R|\omega^R_{k_1}. \quad (7)$$

In order to lighten the notations, we leave out the exponents n in this section.
For any primal edge $A_j=[k_1k_2]$ and its associated dual edges $A'_{j,\alpha}$, the fluxes $F_{i_1 j}$ and $F_{k_1 j,\alpha}$ are sums of a diffusive and a convective contribution. The diffusive part is evaluated as in [5] using a gradient defined by two directions, on each triangle $k_1 i_\alpha k_2=:D_{j,\alpha}$ (also called "half-diamond cell"), see Fig. 1 (left):

$$\begin{cases}(\nabla_h c)_{i_\alpha j}\cdot\overrightarrow{i_\alpha\sigma}=c_\sigma-c_{i_\alpha}\\(\nabla_h c)_{i_\alpha j}\cdot\overrightarrow{k_1k_2}=c_{k_2}-c_{k_1}\end{cases}, \quad (8)$$

where σ is the midpoint of A_j. Formulas (8) are equivalent to

$$(\nabla_h c)_{i_\alpha j}=\frac{1}{|D_{j,\alpha}|}\Big((c_{k_2}-c_{k_1})|A'_{j,\alpha}|\boldsymbol{n}'_{k_1 j,\alpha}+(c_\sigma-c_{i_\alpha})|A_j|\boldsymbol{n}_{i_1 j}\Big), \quad (9)$$

where $\boldsymbol{n}_{i_1 j}$ is the outward normal to T_{i_1} on A_j and $\boldsymbol{n}'_{k_1 j,\alpha}$ the outward normal to P_{k_1} on $A'_{j,\alpha}$. The unknown c_σ is introduced both to deal with possibly discontinuous tensors $\boldsymbol{K}$ and to be able to write a local discretization adapted to domain decomposition, as will be shown in Section 3. The gradient $(\nabla_h c)_{i_\alpha j}$ is used in the diffusive part of $F_{i_\alpha j}$ and in the diffusive part of $F_{k_1 j,\alpha}$ and $F_{k_2 j,\alpha}$. Let us denote by $[a]^+$ and $[a]^-$ the

positive and negative part of a such that $a = [a]^- + [a]^+$. The convective part of the flux on the primal mesh is discretized with an upwind scheme which is local to the half-diamond cell $D_{j,\alpha}$:

$$(\boldsymbol{b}c\cdot\boldsymbol{n})_{i_\alpha j} := [(\boldsymbol{b}\cdot\boldsymbol{n})_{i_\alpha j}]^+ c_{i_\alpha} + [(\boldsymbol{b}\cdot\boldsymbol{n})_{i_\alpha j}]^- c_\sigma. \tag{10}$$

This upwinding using c_σ ensures that the discretization of the convection flux is local to a subdomain. This is the idea borrowed from [6]. On the dual mesh, we use a standard upwind scheme:

$$(\boldsymbol{b}c\cdot\boldsymbol{n}^{'})_{k_1 j,\alpha} := [\boldsymbol{b}_{j,\alpha}\cdot\boldsymbol{n}^{'}_{k_1 j,\alpha}]^+ c_{k_1} + [\boldsymbol{b}_{j,\alpha}\cdot\boldsymbol{n}^{'}_{k_1 j,\alpha}]^- c_{k_2}. \tag{11}$$

In (10), $(\boldsymbol{b}\cdot\boldsymbol{n})_{i_\alpha j}$ is defined by (recall that $\boldsymbol{b}\cdot\boldsymbol{n}$ is continuous through primal edges)

$$(\boldsymbol{b}\cdot\boldsymbol{n})_{i_\alpha j} := \frac{1}{|A_j|}\int_{A_j} \boldsymbol{b}\cdot\boldsymbol{n}_{i_\alpha j}(\xi)d\xi. \tag{12}$$

In (11), $\boldsymbol{b}_{j,\alpha}$ is the mean-value of $\boldsymbol{b}$ over $A^{'}_{j,\alpha}$. The fluxes are then defined as follows:

$$F_{i_\alpha j} := [\boldsymbol{K}_{i_\alpha j}(\nabla_h c)_{i_\alpha j}]\cdot\boldsymbol{n}_{i_\alpha j} - (\boldsymbol{b}c\cdot\boldsymbol{n})_{i_\alpha j}, \tag{13}$$

$$F_{k_1 j,\alpha} := [\boldsymbol{K}_{j,\alpha}(\nabla_h c)_{i_\alpha j}]\cdot\boldsymbol{n}^{'}_{k_1 j,\alpha} - (\boldsymbol{b}c\cdot\boldsymbol{n}^{'})_{k_1 j,\alpha}. \tag{14}$$

In (13) and (14), $\boldsymbol{K}_{i_\alpha j}$ and $\boldsymbol{K}_{j,\alpha}$ are the mean-values of $\boldsymbol{K}_{|T_{i_\alpha}}$ over A_j and $A^{'}_{j,\alpha}$, respectively (we recall that $\boldsymbol{K}$ may be discontinuous through primal edges A_j). In order to complete the definition of the scheme, we still need an equation for each c_σ, and one equation for each boundary dual cell. If σ is not on $\partial\Omega$, c_σ is eliminated by requiring the flux conservation through the common interface $\partial T_{i_1}\cap\partial T_{i_2}$:

$$F_{i_1 j} + F_{i_2 j} = 0. \tag{15}$$

Formula (15) defines a unique c_σ that we replace in (9) and (10). For nodes σ and k located on the Dirichlet boundary, we set

$$c_\sigma = c_k = 0 \ , \ \forall\sigma\in\partial\Omega \ , \ \forall k\in\partial\Omega. \tag{16}$$

Theorem 1. *We suppose that $\nabla\cdot\boldsymbol{b}\geq 0$ and that $\boldsymbol{K}$ is a bounded, uniformly definite positive matrix. Then, the discrete convection-diffusion problem in the global domain Ω, defined by formulas (5) to (16) is well-posed.*

3 The multidomain DDFV scheme

In this part we describe the local DDFV scheme in a subdomain together with the discretization of the Robin conditions (2)–(3).

The subdomain scheme is not modified for primal cells : we still use (5) and (13) with the superscript ^{L} (resp. ^{R}) for Ω_L (resp. Ω_R), $c_\sigma^{L,R}=0$ on the Dirichlet boundary and (15) when σ is not on $\partial\Omega$ nor on Γ. Moreover, when σ, midpoint of a primal edge A_j, is on Γ, we discretize the Robin conditions (2)–(3) on A_j by

$$F_{i_1 j}^{L,n}+\lambda_{L,j}\,c_\sigma^{L,n}=-F_{i_2 j}^{R,n}+\lambda_{L,j}\,c_\sigma^{R,n}, \tag{17}$$

$$F_{i_2 j}^{R,n}+\lambda_{R,j}\,c_\sigma^{R,n}=-F_{i_1 j}^{L,n}+\lambda_{R,j}\,c_\sigma^{L,n}, \tag{18}$$

where $\lambda_{L,j}$ and $\lambda_{R,j}$ are discrete counterparts of λ_L and λ_R defined on each primal edge A_j. In (17) and (18), we use the convention that i_1 is in Ω_L and i_2 in Ω_R. We remark that (17)–(18) are equivalent to $c_\sigma^{R,n}=c_\sigma^{L,n}$ and $F_{i_1 j}^{L,n}+F_{i_2 j}^{R,n}=0$.

On interior dual cells, the scheme is not modified: we still use (6) with the superscript ^{L} (resp. ^{R}) for Ω_L (resp. Ω_R). Moreover, $c_k^{L,R}=0$ if k is a node located on the Dirichlet boundary. Finally, if k_1 belongs to $\Gamma\setminus\partial\Omega$, then we denote by $P_{k_1}^L$ (resp. $P_{k_1}^R$) the boundary dual cell in Ω_L (resp. Ω_R) to which k_1 is associated (see Fig. 1, right). The cell $P_{k_1}^L$ (resp. $P_{k_1}^R$) has two types of edges: the edges $A'_{j,\alpha}$ that belong to $\partial P_{k_1}^L\setminus\Gamma$ (resp. $\partial P_{k_1}^R\setminus\Gamma$) and the edges on $\partial P_{k_1}^L\cap\Gamma$ (resp. $\partial P_{k_1}^R\cap\Gamma$). Integrating (1) on $P_{k_1}^L$ and over I_n yields the approximation

$$\omega_{k_1}^L|P_{k_1}^L|\left(\frac{c_{k_1}^{Ln}-c_{k_1}^{Ln-1}}{\Delta t_n}\right)-\sum_{A'_{j,\alpha}\subset\partial P_{k_1}^L}|A'_{j,\alpha}|F_{k_1 j,\alpha}^n-|\partial P_{k_1}^L\cap\Gamma|F_{k_1,\Gamma}^{Ln}=|P_{k_1}^L|\,f_{k_1}^{Ln}, \tag{19}$$

where $F_{k_1,\Gamma}^{Ln}$ is an approximation of $\frac{1}{\Delta t_n|\partial P_{k_1}^L\cap\Gamma|}\int_{I_n}\int_{\partial P_{k_1}^L\cap\Gamma}(\boldsymbol{K}\nabla c-\boldsymbol{b}c)\cdot\boldsymbol{n}_L$ and $f_{k_1}^{Ln}$ is defined similarly to $f_{k_1}^n$ in (6) in which P_{k_1} is replaced by $P_{k_1}^L$. In the same way, we define $F_{k_1,\Gamma}^{Rn}$ and $f_{k_1}^{Rn}$, and we obtain the following approximation of (1) on $P_{k_1}^R$

$$\omega_{k_1}^R|P_{k_1}^R|\left(\frac{c_{k_1}^{Rn}-c_{k_1}^{Rn-1}}{\Delta t_n}\right)-\sum_{A'_{j,\alpha}\subset\partial P_{k_1}^R}|A'_{j,\alpha}|F_{k_1 j,\alpha}^n-|\partial P_{k_1}^R\cap\Gamma|F_{k_1,\Gamma}^{Rn}=|P_{k_1}^R|\,f_{k_1}^{Rn}. \tag{20}$$

Equations (19) and (20) introduce new flux unknowns $F_{k_1,\Gamma}^{Ln}$ and $F_{k_1,\Gamma}^{Rn}$ which are related to the boundary unknowns $c_{k_1}^{Ln}$ and $c_{k_1}^{Rn}$ by the following dual approximations of the Robin boundary conditions (2)–(3)

$$F_{k_1,\Gamma}^{Ln}+\lambda_{L,k_1}c_{k_1}^{Ln}=-F_{k_1,\Gamma}^{Rn}+\lambda_{L,k_1}c_{k_1}^{Rn}, \tag{21}$$

$$F_{k_1,\Gamma}^{Rn}+\lambda_{R,k_1}c_{k_1}^{Rn}=-F_{k_1,\Gamma}^{Ln}+\lambda_{R,k_1}c_{k_1}^{Ln}, \tag{22}$$

where λ_{L,k_1} and λ_{R,k_1} are discrete counterparts of λ_L and λ_R defined on each dual intersection $\partial P_{k_1}^L\cap\Gamma=\partial P_{k_1}^R\cap\Gamma$. We remark that (21) and (22) are equivalent to $c_{k_1}^{Ln}=c_{k_1}^{Rn}$ and $F_{k_1,\Gamma}^{Ln}+F_{k_1,\Gamma}^{Rn}=0$. With these equalities for all time steps, adding (19) and (20) and using (7) yields (6) on $P_{k_1}=P_{k_1}^L\cup P_{k_1}^R$, the inner dual cell of the global domain Ω.

In order to study the well-posedness of the subdomain problems, we restrict ourselves to one subdomain, e.g. Ω_L. Recalling that $(\boldsymbol{b}\cdot\boldsymbol{n})_{i_\alpha j}$ is defined by (12) and defining $(\boldsymbol{b}\cdot\boldsymbol{n})^L_{k_1}$ by

$$(\boldsymbol{b}\cdot\boldsymbol{n})^L_{k_1,\Gamma} := \frac{1}{|\partial P^L_{k_1}\cap\Gamma|}\int_{\partial P^L_{k_1}\cap\Gamma}\boldsymbol{b}\cdot\boldsymbol{n}_L(\xi)d\xi,$$

we can prove the following theorem

Theorem 2. *Under the hypothesis of Theorem 1, if $\lambda_{L,j} > \frac{1}{2}(\boldsymbol{b}\cdot\boldsymbol{n})_{i_1 j}$ for all j such that $A_j \subset \Gamma$ and if $\lambda_{L,k_1} > \frac{1}{2}(\boldsymbol{b}\cdot\boldsymbol{n})^L_{k_1,\Gamma}$ for all k such that $\partial P^L_k \cap \Gamma \neq \emptyset$, then the discrete problem in Ω_L, defined by formulas (5)-(6) and (13) to (16) with the superscript ^{L}, formula (19) for boundary dual cells, and the Robin conditions*

$$\begin{aligned} F^{L,n}_{i_1 j} + \lambda_{L,j}\, c^{L,n}_\sigma &= g^{L,n}_j \quad (\textit{on primal edges } A_j \subset \Gamma)\\ F^{Ln}_{k_1,\Gamma} + \lambda_{L,k_1} c^{Ln}_{k_1} &= g^{L,n}_{k_1} \quad (\textit{on dual edges } \partial P^L_{k_1}\cap\Gamma), \end{aligned}$$

with $g^{L,n}_j$ and $g^{L,n}_{k_1}$ given real numbers, is well-posed.

4 The Schwarz algorithm

Let S denote the superscript ^{L} or ^{R}. The discrete Schwarz algorithm is defined as follows: let $(c^{Sn(\ell)}_i, c^{Sn(\ell)}_k, c^{Sn(\ell)}_\sigma)$ and $(F^{Sn(\ell)}_{ij}, F^{Sn(\ell)}_{kj,\alpha}, F^{Sn(\ell)}_{k,\Gamma})$ be given approximations, at step ℓ, of c at nodes i, k, σ and $(\boldsymbol{K}\nabla c - \boldsymbol{b}c)\cdot\boldsymbol{n}$ at edges $A_j, A'_{j,\alpha}, \partial P^S_k \cap \Gamma$. Then we compute $(c^{Sn(\ell+1)}_i, c^{Sn(\ell+1)}_k, c^{Sn(\ell+1)}_\sigma)$ and $(F^{Sn(\ell+1)}_{ij}, F^{Sn(\ell+1)}_{kj,\alpha}, F^{Sn(\ell+1)}_{k,\Gamma})$ as the solution of (5)-(6) and (13) to (16) with the superscript ^{L} (resp. ^{R}), formula (19) (resp. (20)) and the following Robin conditions for interface primal and dual cells:

$$\begin{aligned} F^{Ln(\ell+1)}_{i_1 j} + \lambda_{L,j}\, c^{Ln(\ell+1)}_\sigma &= -F^{Rn(\ell)}_{i_2 j} + \lambda_{L,j}\, c^{Rn(\ell)}_\sigma,\\ F^{Ln(\ell+1)}_{k_1,\Gamma} + \lambda_{L,k_1} c^{Ln(\ell+1)}_{k_1} &= -F^{Rn(\ell)}_{k_1,\Gamma} + \lambda_{L,k_1} c^{Rn(\ell)}_{k_1},\\ F^{Rn(\ell+1)}_{i_2 j} + \lambda_{R,j}\, c^{Rn(\ell+1)}_\sigma &= -F^{Ln(\ell)}_{i_1 j} + \lambda_{R,j}\, c^{Ln(\ell)}_\sigma,\\ F^{Rn(\ell+1)}_{k_1,\Gamma} + \lambda_{R,k_1} c^{Rn(\ell+1)}_{k_1} &= -F^{Ln(\ell)}_{k_1,\Gamma} + \lambda_{R,k_1} c^{Ln(\ell)}_{k_1}. \end{aligned}$$

Theorem 3. *Under the hypothesis of Theorem 2, if $\lambda_{R,k_1} - \lambda_{L,k_1} - (\boldsymbol{b}\cdot\boldsymbol{n})^L_{k_1,\Gamma} = 0$ for all k such that $\partial P^L_k \cap \Gamma \neq \emptyset$ and if $\lambda_{R,j} - \lambda_{L,j} - (\boldsymbol{b}\cdot\boldsymbol{n})_{i_1 j} = 0$ for all j such that $A_j \subset \Gamma$, then the discrete Schwarz algorithm converges to the solution of the discrete convection-diffusion problem in the domain Ω, defined by formulas (5) to (16).*

Remark 1. Following [8, 9], the Robin parameters are chosen in the form

$$\lambda_{L,j} = (-(\boldsymbol{b}\cdot\boldsymbol{n})_{i_1 j} + p_{L,j})/2 \ , \ \lambda_{R,j} = ((\boldsymbol{b}\cdot\boldsymbol{n})_{i_1 j} + p_{R,j})/2, \tag{23}$$

$$\lambda_{L,k_1} = (-(\boldsymbol{b}\cdot\boldsymbol{n})^L_{k_1,\Gamma} + p_{L,k_1})/2 \ , \ \lambda_{R,k_1} = ((\boldsymbol{b}\cdot\boldsymbol{n})^L_{k_1,\Gamma} + p_{R,k_1})/2, \tag{24}$$

where $p_{L,j}$, $p_{R,j}$ and p_{L,k_1}, p_{R,k_1} are the primal and dual parameters which optimize the convergence factor of the continuous algorithm (4). This optimization is performed by a numerical minimization process. With the form given by (23)-(24), the hypothesis in Theorem 3 reduces to $p_{L,j} = p_{R,j}$ and $p_{L,k_1} = p_{R,k_1}$.

Remark 2. The scheme we proposed here is different from the one developed in [1]. On the other hand, it is shown independently in [7], using an analysis of the convergence factor at the discrete level, that our method leads to a faster convergence than the approach in [1]. In our simulations, we observed that using the optimized parameters at the discrete level does not improve significantly the convergence.

5 Numerical Results

Here, the Robin parameter for $\Omega_{L/R}$ is taken as the mean value of all $\lambda_{L/R,j}$ and $\lambda_{L/R,k_1}$ and is denoted $\lambda^*_{L/R}$. Moreover, $\boldsymbol{b}\cdot\boldsymbol{n} = 0$ on Γ in our tests, thus $\lambda^*_{L/R} = p^*$, the same value for all primal and dual (L and R) interface cells. Its discrete counterpart p^*_h is obtainded in the same way but with an optimization of the discrete convergence factor, denoted ρ_h. We assume that $\boldsymbol{K} = \nu\boldsymbol{I}$ where $\boldsymbol{I}$ is the identity matrix.

In the first test case, we take $\Omega_L = (0,2.5)\times(0,5)$ and $\Omega_R = (2.5,5)\times(0,5)$, with $T = 1$, $\omega_L = \omega_R = 1$, $\boldsymbol{b} = \boldsymbol{0}$, $\nu_{|\Omega_L} = 0.06$, and $\nu_{|\Omega_R} = 1$. The mesh size and time step are $h = \frac{5}{100}$ and $\Delta t = \frac{1}{70}$ respectively. On Fig. 2 we show a section along the diagonal $(w_m,k_m)-(w_M,k_M)$ of ρ_h (top left), where $(w_m,w_M)\times(k_m,k_M)$ is the frequencies interval over which ρ_h is optimized, with $w_m = \frac{\pi}{T}$, $w_M = \frac{\pi}{\Delta t}$, $k_m = \frac{\pi}{5}$, $k_M = \frac{\pi}{h}$, and the error versus the number of iterations for the Schwarz algorithm (top right) with p^* and p^*_h. We simulate directly the error equations, $f = 0$ and use a random initial guess so that all the frequency components are present. We observe that using p^*_h or p^* give similar results. We also observe the equioscillation property [2] with p^*_h.

In the second test case, we take $\Omega_L = (0,0.5)\times(0,1)$ and $\Omega_R = (0.5,1)\times(0,1)$, with $T = 1$, $\omega_L = 0.2$, $\omega_R = 1$, $\nu_{|\Omega_L} = 0.005$, $\nu_{|\Omega_R} = 0.01$, and a rotating velocity field $\boldsymbol{b} = (-sin(\pi(y-\frac{1}{2}))cos(\pi(x-\frac{1}{2})), cos(\pi(y-\frac{1}{2}))sin(\pi(x-\frac{1}{2})))$. We take $h = \frac{1}{100}$ and $\Delta t = \frac{1}{50}$. On Fig. 2 we show the computed solution at time $t = 0.4$ (bottom left) and the error versus the number of iterations (bottom right) for different values of the Robin parameter p, taken constant along the interface. We take $f = 0$ and a random initial guess. We observe that p^* is close to the optimal numerical value.

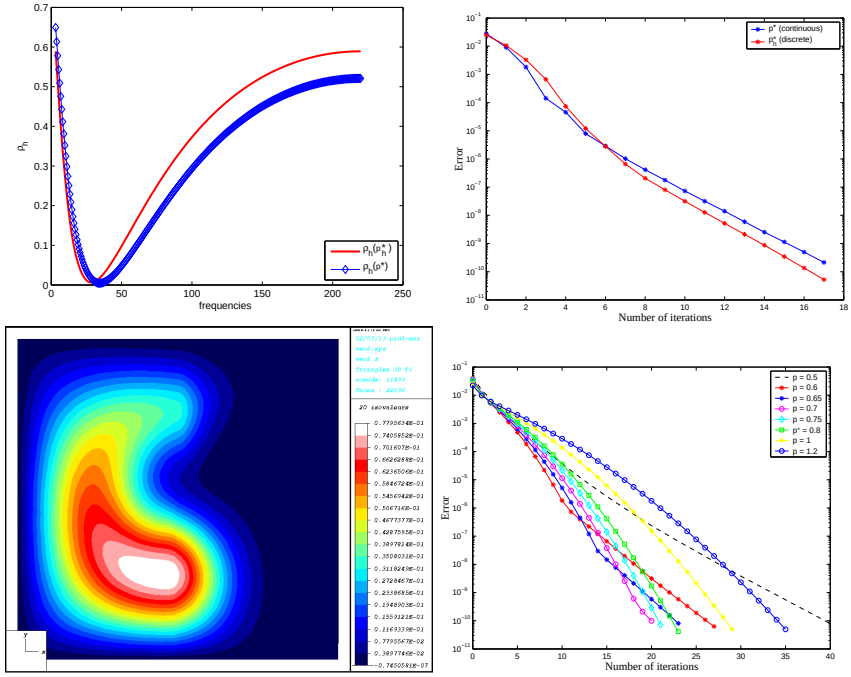


Fig. 2 Top: Discrete convergence factor (left) and error versus iterations (right), with p^* and p_h^*. Bottom: solution at time $t = 0.4$ (left) and error versus iterations (right) for different values of p.

References

1. Boyer, F., Hubert, F., Krell, S.: Non-overlapping Schwarz algorithm for solving 2D m-DDFV schemes. IMA Journal on Numerical Analysis **4 (30)**, 1062–1100 (2010)
2. Bennequin, D., Gander, M.J., Halpern, L.: A Homographic Best Approximation Problem with Application to Optimized Schwarz Waveform Relaxation. Math. of Comp. **78**, 185–223 (2009)
3. Martin, V.: An optimized Schwarz waveform relaxation method for the unsteady convection diffusion equation in two dimensions. Appl. Numer. Math. **52**, 401–428 (2005)
4. Halpern, L., Japhet, C., Szeftel, J.: Optimized Schwarz waveform relaxation and discontinuous Galerkin time stepping for heterogeneous problems. SIAM J. Numer. Anal. **50 (5)**, 2588–2611 (2012)
5. Domelevo, K., Omnes, P.: A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids. Math. Model. Numer. Anal. **39**, 1203–1249" (2005)
6. Haeberlein, F.: Time Space Domain Decomposition Methods for Reactive Transport - Application to CO2 Geological Storage. PhD thesis, University Paris XIII (2011)
7. Gander, M.J., Hubert, F., Krell, S.: Optimized Schwarz algorithms in the framework of DDFV schemes. In this proceedings.
8. Gander, M.J., Halpern, L., Kern, M.: A Schwarz waveform relaxation method for advection-diffusion-reaction problems with discontinuous coefficients and non-matching grids. Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering **55**, 283–290. Springer (2007)
9. Halpern, L., Japhet, C., Omnes, P.: Nonconforming in time domain decomposition method for porous media applications. ECCOMAS CFD 2010, C.F. Pereira, A.Sequeira (Eds) (2010)

Block Jacobi relaxation for plane wave discontinuous Galerkin methods

T. Betcke[1], M.J Gander[2], and J. Phillips[1]

1 Introduction

Nonpolynomial finite element methods for Helmholtz problems have seen much attention in recent years in the engineering and mathematics community. The idea is to use instead of standard polynomials Trefftz-type basis functions that already satisfy the Helmholtz equation, such as plane waves [17], Fourier-Bessel functions [8] or fundamental solutions [4]. To approximate the inter-element interface conditions between elements several possibilities exist, such as the ultra-weak variational formulation (UWVF [6]), plane wave discontinuous Galerkin methods (PWDG [15]), partition of unity finite elements (PUFEM [3]), least-squares methods [18, 5], or Lagrange-multiplier approaches [10].

The advantage of Trefftz methods is that they often require fewer degrees of freedom than standard polynomial finite element methods since the basis functions already oscillate with the correct wavenumber. The disadvantage is that the resulting linear systems are often significantly ill-conditioned, making direct solvers or efficient preconditioning for iterative solvers necessary. For very large problems, especially in three dimensions, direct solvers become prohibitively expensive, and preconditioning iterative solvers is a difficult problem for the Helmholtz equation as demonstrated in [9].

Domain decomposition methods, in particular optimized Schwarz methods, have proven to still be effective iterative solvers for finite elements and discontinuous Galerkin methods with polynomial basis functions; for the Helmholtz equation, see [11, 12], and for Maxwell's equation, see [1, 7].

In this paper we consider block Jacobi relaxation methods for the PWDG method. In the classical finite element case a block Jacobi relaxation is equivalent to a classical Schwarz method with Dirichlet transmission conditions, see for example [13]. This is however not necessarily the case for discontinuous Galerkin methods, see [14]. We investigate in this short paper what kind of domain decomposition methods one obtains when simply performing a block Jacobi relaxation in a PWDG discretization of the Helmholtz equation, and also show how one can obtain optimized Schwarz methods for such discretizations. Motivated by the block Jacobi relaxation we present a simple algebraic decomposition approach of the system matrix in PWDG methods and demonstrate for an example problem with plane wave basis functions its performance for iterative solvers.

[1]Department of Mathematics, University College London, UK, e-mail: t.betcke@ucl.ac.uk, joel.m.phillips@gmail.com .[2] Section of Mathematics, University of Geneva, Switzerland, martin.gander@unige.ch

While in this paper we focus on plane wave basis functions the results are certainly more generally applicable for other Trefftz basis functions, and also for standard polynomial basis functions.

We consider the following model problem: find $u \in \mathscr{C}^2(\Omega) \cap H^1(\Omega)$, such that

$$-\Delta u - k^2 u = f \quad \text{in } \Omega, \qquad \frac{\partial u}{\partial n} - Su = g \quad \text{on } \partial\Omega. \tag{1}$$

Here, $\Omega \subset \mathbb{R}^d$, $d = 2,3$, is a bounded domain with Lipschitz boundary $\Gamma := \partial\Omega$ and $g \in H^{-1/2}(\Gamma)$. The operator S is often an exact or approximate Dirichlet to Neumann (DtN) map, e.g. $S = ik$.

We will use the following notation: the triangulation into finite elements of maximum diameter h is denoted by $\mathscr{T}_h$. Let $K \in \mathscr{T}_h$ be an element of the triangulation. The outward normal direction to K is denoted by n. On an edge e between two elements K^- and K^+ we define for a scalar quantity v the jumps $[\![v]\!] := v^- n^- + v^+ n^+$ and averages $\{\{v\}\} := \frac{1}{2}(v^- + v^+)$. Similarly, for a vector quantity σ we define $[\![\sigma]\!] := \sigma^- \cdot n^- + \sigma^+ \cdot n^+$ and $\{\{\sigma\}\} := \frac{1}{2}(\sigma^- + \sigma^+)$. On boundary edges we define $[\![v]\!] = v\mathbf{n}$ and $\{\{\sigma\}\} = \sigma$.

The set of all interior edges is denoted by $\mathscr{E}^{(int)}$ and the set of all eges is denoted by $\mathscr{E}$. Also, let $\tilde{\Omega}$ be defined by $\tilde{\Omega} := \bigcup_{K \in \mathscr{T}_h} K$.

2 Plane Wave Discontinuous Galerkin Methods

In the following we give a brief overview of the Plane Wave Discontinuous Galerkin Method (PWDG). For a more detailed introduction and convergence results see [15, 16]. For each element $K_i \in \mathscr{T}_h$ we define a local approximation space $V_i := \text{span}\{\Psi_1^{(i)}, \ldots, \Psi_{N_i}^{(i)}\}$, where $\Psi_\ell^{(i)} \in \mathscr{C}^2(K_i) \cap H^1(K_i)$ and satisfies $\Delta\Psi_\ell^{(i)} + k^2\Psi_\ell^{(i)} = 0$, $\ell = 1, \ldots, N_i$. A frequent choice is the plane wave basis set $PW_i^{(N_i)}$ defined by $\Psi_\ell^{(i)}(x) := e^{ikd_\ell \cdot x}$, where the d_ℓ are direction vectors with $\|d_\ell\|_2 = 1_2$. In two dimensions, typically $d_\ell = \frac{2\pi(\ell-1)}{N_i}$, that is we take equally spaced directions on the unit circle. In three dimensions several possibilities exist to choose approximately equally spaced directions on the unit sphere (see e.g. [17]). By $V := \{v \in L^2(\Omega) : \ v|_{K_i} \in V_i \ \forall K_i \in \mathscr{T}_h\}$ we denote the global approximation space.

Let $K \subset \mathscr{T}_h$. By multiplying (1) with a test function $v \subset V$ on K and integrating by parts we obtain

$$\int_K \nabla u \cdot \overline{\nabla v} dV - k^2 \int_K u\bar{v} dV - \int_{\partial K} \nabla u \cdot n\bar{v} dS = \int_K f\bar{v} dV.$$

A further integration by parts yields

$$\int_K \overline{(-\Delta v - k^2 v)} u + \int_{\partial K} u \cdot \overline{\nabla v \cdot n} dS - \int_{\partial K} \nabla u \cdot n\bar{v} dS = \int_K f\bar{v} dV.$$

Define $\sigma := \frac{1}{ik}\nabla u$ and note that $-\Delta v - k^2 v = 0$. It follows that

$$\int_{\partial K} u \cdot \overline{\nabla v \cdot n} dS - ik \int_{\partial K} \sigma \cdot n \bar{v} dS = \int_K f \bar{v} dV.$$

Using the DG summation formula, see [2],

$$\sum_{K \in \mathscr{T}_h} \int_{\partial K} q_K \phi_K \cdot n_K = \int_{\mathscr{E}} [\![q]\!] \cdot \{\!\{\phi\}\!\} dS + \int_{\mathscr{E}^{(int)}} \{\!\{q\}\!\} [\![\phi]\!] dS,$$

where q is a scalar and ϕ a vector quantity we obtain

$$\begin{aligned} \int_{\mathscr{E}} [\![u]\!] \cdot \left\{\!\!\left\{\overline{\nabla v}\right\}\!\!\right\} dS + \int_{\mathscr{E}^{(int)}} \{\!\{u\}\!\} \left[\!\!\left[\overline{\nabla v}\right]\!\!\right] dS - ik \int_{\mathscr{E}} \{\!\{\sigma\}\!\} \cdot [\![\bar{v}]\!] dS \\ - ik \int_{\mathscr{E}^{(int)}} [\![\sigma]\!] \{\!\{\bar{v}\}\!\} dS = \int_{\tilde{\Omega}} f \bar{v} dV. \end{aligned} \tag{2}$$

We now approximate u and σ on the edges in terms of their numerical fluxes $\hat{u}_h$ and $\hat{\sigma}_h$, defined by

$$\hat{\sigma}_h := \frac{1}{ik} \{\!\{\nabla_h u_h\}\!\} - \alpha [\![u_h]\!] - \frac{\tau}{ik} [\![\nabla_h u_h]\!], \quad \hat{u}_h := \{\!\{u_h\}\!\} + \tau \cdot [\![u_h]\!] - \frac{\beta}{ik} [\![\nabla_h u_h]\!] \tag{3}$$

for interior edges, and by

$$\hat{\sigma}_h := \frac{1}{ik} \nabla_h u_h - \frac{(1-\delta)}{ik} (\nabla_h u_h - S u_h \mathbf{n} - g\mathbf{n}), \; \hat{u}_h := u_h - \frac{\delta}{ik} (\nabla_h u_h \cdot \mathbf{n} - S u_h - g) \tag{4}$$

for boundary edges. Choices for the parameters α, β, τ and δ are discussed in [15]. In particular, it is shown there that with the choice $\alpha = \beta = \delta = 0.5$, $\tau = 0$ the PWDG is equivalent to the UWVF. By replacing u and σ in (2) with their corresponding numerical fluxes, noting that

$$\{\!\{\hat{u}_h\}\!\} = \hat{u}_h, \quad \{\!\{\hat{\sigma}_h\}\!\} = \hat{\sigma}_h, \quad [\![\hat{u}_h]\!] = [\![\hat{\sigma}_h]\!] = 0,$$

on interior edges and using $[\![\hat{u}_h]\!] = \hat{u}_h \mathbf{n}$, $\{\!\{\sigma_h\}\!\} = \hat{\sigma}_h$ on boundary edges we arrive at the following variational problem: find $u_h \in V$, such that

$$a(u_h, v_h) = \ell(v_h) - b(g, v_h) \quad \forall v_h \in V, \tag{5}$$

where

$$\begin{aligned} a(u_h, v_h) := & \int_{\mathscr{E}^{(ext)}} u_h \overline{\nabla_h v_h \cdot \mathbf{n}} dS - \frac{\delta}{ik} \int_{\mathscr{E}^{(ext)}} \nabla_h u_h \cdot \mathbf{n} \overline{\nabla_h v_h \cdot \mathbf{n}} dS + \frac{\delta}{ik} \int_{\mathscr{E}^{(ext)}} S u_h \overline{\nabla_h v_h \cdot \mathbf{n}} dS \\ & + \int_{\mathscr{E}^{(int)}} \{\!\{u_h\}\!\} \left[\!\!\left[\overline{\nabla_h v_h}\right]\!\!\right] dS + \int_{\mathscr{E}^{(int)}} \tau \cdot [\![u_h]\!] \left[\!\!\left[\overline{\nabla_h v_h}\right]\!\!\right] dS - \frac{\beta}{ik} \int_{\mathscr{E}^{(int)}} [\![\nabla_h u_h]\!] \left[\!\!\left[\overline{\nabla_h v_h}\right]\!\!\right] dS \\ & - \delta \int_{\mathscr{E}^{(ext)}} \nabla_h u_h \cdot \mathbf{n} \overline{v_h} dS - (1-\delta) \int_{\mathscr{E}^{(ext)}} S u_h \overline{v_h} dS - \int_{\mathscr{E}^{(int)}} \{\!\{\nabla_h u_h\}\!\} \cdot [\![\bar{v}]\!] dS \end{aligned}$$

$$+\ \alpha ik \int_{\mathscr{E}^{(int)}} [\![u_h]\!] \cdot [\![\overline{v_h}]\!]\, dS + \int_{\mathscr{E}^{(int)}} [\![\nabla_h u_h]\!]\, \tau \cdot [\![\overline{v_h}]\!]\, dS,$$

$$b(g, v_h) := \frac{\delta}{ik} \int_{\mathscr{E}^{(ext)}} g \overline{\nabla v_h \cdot \mathbf{n}} dS - (1-\delta) \int_{\mathscr{E}^{(ext)}} g \overline{v_h} dS,$$

$$\ell(v_h) := \int_{\tilde{\Omega}} f \overline{v_h} dV.$$

3 A natural Schwarz iteration for the UWVF

In this section we show that a simple block relaxation of the UWVF gives rise to a Schwarz algorithm with Robin transmission conditions, and not the classical Schwarz algorithm with Dirichlet transmission conditions. We consider a simple example problem of a domain Ω decomposed into two subdomains Ω_1 and Ω_2 with interface $\Gamma_{12} = \overline{\Omega_1} \cap \overline{\Omega_2}$. We start by defining the following optimized Schwarz iteration with Robin transmission conditions and optimization parameter p:

$$\begin{aligned}
-\Delta u_1^{(n+1)} - k^2 u_1^{(n+1)} &= f && \text{in } \Omega_1,\\
-\Delta u_2^{(n+1)} - k^2 u_2^{(n+1)} &= f && \text{in } \Omega_2,\\
\tfrac{\partial u_1^{(n+1)}}{\partial n_1} + p u_1^{(n+1)} &= \tfrac{\partial u_2^{(n)}}{\partial n_1} + p u_2^{(n)} && \text{on } \Gamma_{12},\\
\tfrac{\partial u_2^{(n+1)}}{\partial n_2} + p u_2^{(n+1)} &= \tfrac{\partial u_1^{(n)}}{\partial n_2} + p u_1^{(n)} && \text{on } \Gamma_{12},\\
\tfrac{\partial u_1^{(n+1)}}{\partial n_1} + ik u_1^{(n+1)} &= g && \text{on } \Gamma \cap \partial\Omega_1,\\
\tfrac{\partial u_2^{(n+1)}}{\partial n_2} + ik u_2^{(n+1)} &= g && \text{on } \Gamma \cap \partial\Omega_2.
\end{aligned} \tag{6}$$

Discretizing each of the subproblems with the PWDG and UWVF flux parameters, and setting $p = ik$ gives the sequence of discrete equations

$$a_1(u_{h,1}^{(n+1)}, v_h) = \ell_1(v_h) - b_{\Gamma \cap \partial\Omega_1}(g, v_h) - b_{\Gamma_{12}}\left(\frac{\partial u_2^{(n)}}{\partial n_1} + ik u_2^{(n)}, v_h\right),\ v_h \in V_1^{(h)},$$

$$a_2(u_{h,2}^{(n+1)}, v_h) = \ell_1(v_h) - b_{\Gamma \cap \partial\Omega_2}(g, v_h) - b_{\Gamma_{21}}\left(\frac{\partial u_1^{(n)}}{\partial n_2} + ik u_1^{(n)}, v_h\right),\ v_h \in V_2^{(h)}.$$

Theorem 1. *A classical block-Jacobi relaxation applied to the global variational problem (5) discretized with PWDG and UWVF flux parameters, i.e. setting*

$$\hat{\sigma}_1^{n+1} \cdot \mathbf{n}_1 = \frac{1}{ik} \{\{\nabla u\}\}^{n+1,n} \cdot \mathbf{n}_1 - \frac{1}{2} [\![u]\!]^{n+1,n} \cdot \mathbf{n}_1, \tag{7}$$

$$\hat{\sigma}_2^{n+1} \cdot \mathbf{n}_2 = \frac{1}{ik} \{\{\nabla u\}\}^{n+1,n} \cdot \mathbf{n}_2 - \frac{1}{2} [\![u]\!]^{n+1,n} \cdot \mathbf{n}_2, \tag{8}$$

$$\hat{u}_1^{n+1} = \{\{u\}\}^{n+1,n} - \frac{1}{2ik} [\![\nabla u]\!]^{n+1,n}, \tag{9}$$

$$\hat{u}_2^{n+1} = \{\{u\}\}^{n+1,n} - \frac{1}{2ik}[\![\nabla u]\!]^{n+1,n}, \tag{10}$$

where

$$\{\{\nabla u\}\}^{n+1,n} := \frac{1}{2}((\nabla u^-)^{n+1} + (\nabla u^+)^n), \quad [\![u]\!]^{n+1,n} := \frac{1}{2}((u^-)^{n+1}\mathbf{n}^- + (u^+)^n\mathbf{n}^+),$$

leads precisely to the optimized Schwarz method (6) discretized with PWDG and UWVF, provided the optimization parameter is set to $p = ik$.

Proof. The classical Robin condition for the Helmholtz equation in this formulation uses the flux term

$$\hat{\sigma}_1^{n+1} = \frac{1}{ik}\nabla u_1^{n+1} - \frac{1}{ik}(1-\delta)\left(\nabla u_1^{n+1} + iku_1^{n+1}\cdot\mathbf{n}_1 - (\nabla u_2^n + iku_2^n\cdot\mathbf{n}_1)\right),$$

and similarly for the second flux term. We have to show that this is precisely the flux (7) given by natural algebraic relaxation. We calculate

$$\begin{aligned}\hat{\sigma}_1^{n+1}\cdot\mathbf{n}_1 &= \frac{\delta}{ik}\nabla u_1^{n+1}\cdot\mathbf{n}_1 - (1-\delta)u_1^{n+1} + \frac{1-\delta}{ik}\nabla u_2^n\cdot\mathbf{n}_1 + (1-\delta)u_2^n \\ &= \frac{\delta}{ik}[\![\nabla u]\!]^{n+1,n} - (1-\delta)[\![u]\!]^{n+1,n}\cdot\mathbf{n}_1 + \frac{1}{ik}\nabla u_2^n\cdot\mathbf{n}_1\end{aligned}$$

and choosing $\delta = \frac{1}{2}$, and using the relation

$$\nabla u_2^n\cdot\mathbf{n}_1 = \{\{\nabla u\}\}^{n,n}\cdot\mathbf{n}_1 - \frac{1}{2}[\![\nabla u]\!]^{n,n}$$

we obtain

$$\begin{aligned}\hat{\sigma}_1^{n+1}\cdot\mathbf{n}_1 &= \frac{1}{2ik}[\![\nabla u]\!]^{n+1,n} - \frac{1}{2}[\![u]\!]^{n+1,n}\cdot\mathbf{n}_1 + \frac{1}{ik}(\{\{\nabla u\}\}^{n,n}\cdot\mathbf{n}_1 - \frac{1}{2}[\![\nabla u]\!]^{n,n}) \\ &= \frac{1}{ik}\{\{\nabla u\}\}^{n,n}\cdot\mathbf{n}_1 - \frac{1}{2}[\![u]\!]^{n+1,n}\cdot\mathbf{n}_1 + \frac{1}{2ik}[\![\nabla u]\!]^{n+1,n} - \frac{1}{2ik}[\![\nabla u]\!]^{n,n} \\ &= \frac{1}{ik}\{\{\nabla u\}\}^{n+1,n}\cdot\mathbf{n}_1 - \frac{1}{2}[\![u]\!]^{n+1,n}\cdot\mathbf{n}_1\end{aligned}$$

and the proof for $\hat{\sigma}_1^{n+1}$ is complete. The proof for the other flux terms follows along the same lines.

The choice $p = ik$ corresponds to a low frequency approximation of the optimal transmission condition, see for example [11]. Optimized Schwarz methods use however a different value for the complex parameter p, in order to obtain fast geometric convergence of the method [11, 12]. The question is how to modify the natural relaxation in order to obtain an optimized Schwarz method. In the following this is described for the $\hat{\sigma}$-flux parameter. The result for the $\hat{u}$-flux follows similarly.

Theorem 2. *Performing the modified algebraic relaxation*

$$\hat{\sigma}_1^{n+1}\cdot\mathbf{n}_1 = \frac{1}{ik}\left\{\{\nabla u\}\right\}^{n+1,n}\cdot\mathbf{n}_1 - \frac{1}{2}\left[\!\left[u\right]\!\right]^{n+1,n}\cdot\mathbf{n}_1 + \frac{1}{2}(1-\frac{p}{ik})(u_{1,r}^{n+1}-u_2^n), \quad (11)$$

$$\hat{\sigma}_2^{n+1}\cdot\mathbf{n}_2 = \frac{1}{ik}\left\{\{\nabla u\}\right\}^{n+1,n}\cdot\mathbf{n}_2 - \frac{1}{2}\left[\!\left[u\right]\!\right]^{n+1,n}\cdot\mathbf{n}_2 + \frac{1}{2}(1-\frac{p}{ik})(u_{2,l}^{n+1}-u_1^n), \quad (12)$$

where we needed to introduce for subdomain Ω_1 the additional variable $u_{1,r}^{n+1}$ to represent the quantity from the other side of the interface corresponding to u_2, and on Ω_2 the additional variable $u_{2,l}^{n+1}$ which represents the quantity from the other side of the interface corresponding to u_1, we obtain a discretization of the transmission conditions

$$\frac{\partial u_1^{(n+1)}}{\partial n_1} + pu_1^{(n+1)} = \frac{\partial u_2^{(n)}}{\partial n_1} + pu_2^{(n)}, \quad (13)$$

$$\frac{\partial u_2^{(n+1)}}{\partial n_2} + pu_2^{(n+1)} = \frac{\partial u_1^{(n)}}{\partial n_2} + pu_1^{(n)}. \quad (14)$$

Proof. With the new variables before relaxation, we can write the flux at the interface as

$$\hat{\sigma}_1^{n+1} = \frac{1}{ik}\nabla u_1^{n+1} - \frac{1}{ik}(1-\delta)\left(\nabla u_1^{n+1} + iku_1^{n+1}\cdot\mathbf{n}_1 - (\nabla u_{1,r}^{n+1} + iku_{1,r}^{n+1}\cdot\mathbf{n}_1)\right).$$

In order to substitute the Robin condition from the right, we compute from (13) by adding and subtracting the same term

$$\frac{\partial u_{1,r}^{(n+1)}}{\partial n_1} + iku_{1,r}^{(n+1)} = \frac{\partial u_2^{(n)}}{\partial n_1} + pu_2^{(n)} + (ik-p)u_{1,r}^{(n+1)},$$

which we insert into the flux to obtain

$$\begin{aligned}
\hat{\sigma}_1^{n+1}\cdot\mathbf{n}_1 &= \frac{\delta}{ik}\nabla u_1^{n+1}\cdot\mathbf{n}_1 - (1-\delta)u_1^{n+1} + \frac{1-\delta}{ik}\nabla u_2^n\cdot\mathbf{n}_1 + (1-\delta)\frac{p}{ik}u_2^n + (1-\delta)\frac{ik-p}{ik}u_{1,r}^{n+1} \\
&= \frac{\delta}{ik}\left[\!\left[\nabla u\right]\!\right]^{n+1,n} - (1-\delta)\left[\!\left[u\right]\!\right]^{n+1,n}\cdot\mathbf{n}_1 + \frac{1}{ik}\nabla u_2^n\cdot\mathbf{n}_1 + \frac{1}{2}(1-\frac{p}{ik})(u_{1,r}^{n+1}-u_2^n) \\
&= \frac{1}{ik}\left\{\{\nabla u\}\right\}^{n+1,n}\cdot\mathbf{n}_1 - \frac{1}{2}\left[\!\left[u\right]\!\right]^{n+1,n}\cdot\mathbf{n}_1 + \frac{1}{2}(1-\frac{p}{ik})(u_{1,r}^{n+1}-u_2^n),
\end{aligned}$$

where we used the same simplification as in the proof of Theorem 1 to complete the proof for $\hat{\sigma}_1^{n+1}$. The proof for $\hat{\sigma}_2^{n+1}$ is analogous.

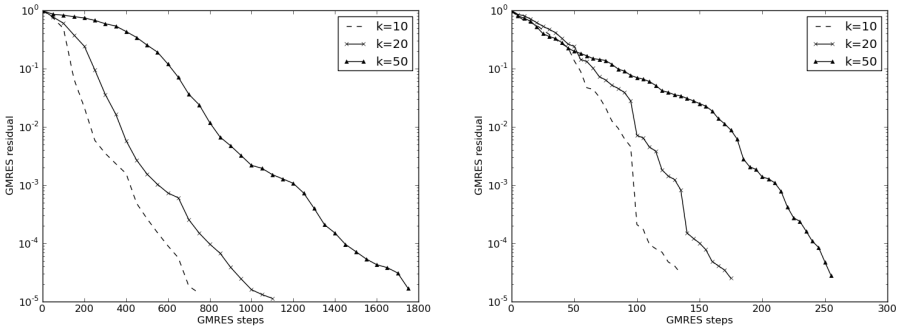


Fig. 1 Left: Convergence of GMRES for the solution of (15) for various wavenumbers k. Right: GMRES convergence for the solution of (16) for various k.

4 Discrete system and numerical results

In this section we present preliminary results for the natural decomposition according to Theorem 1. Results for optimized flux parameters are in preparation. We consider as example a problem partitioned into two subdomains. The global system matrix can be decomposed in the following form.

$$\begin{bmatrix} A_{e_1,e_1} & A_{e_1,e_2} & A_{e_1,i_1} & 0 \\ A_{e_2,e_1} & A_{e_2,e_2} & 0 & A_{e_2,i_2} \\ A_{i_1,e_1} & 0 & A_{i_1,i_1} & 0 \\ 0 & A_{i_2,e_2} & 0 & A_{i_2,i_2} \end{bmatrix} \begin{bmatrix} u_{e_1} \\ u_{e_2} \\ u_{i_1} \\ u_{i_2} \end{bmatrix} = \begin{bmatrix} g_{e_1} \\ g_{e_2} \\ g_{i_1} \\ g_{i_2} \end{bmatrix}. \tag{15}$$

Here, e_1, and e_2 denote degrees of freedom associated with the interface elements from both sides, and i_1 and i_2 denote the interior degrees of freedom. Assume that a fast direct solver is available on each subdomain. Eliminating interior degrees of freedom we arrive at the Schur complement system

$$\begin{bmatrix} A_{e_1,e_1} - A_{e_1,i_1}A_{i_1,i_1}^{-1}A_{i_1,e_1} & A_{e_1,e_2} \\ A_{e_2,e_1} & A_{e_2,e_2} - A_{e_2,i_2}A_{i_2,i_2}^{-1}A_{i_2,e_2} \end{bmatrix} \begin{bmatrix} u_{e_1} \\ u_{e_2} \end{bmatrix} = \begin{bmatrix} g_{e_1} \\ g_{e_2} \end{bmatrix} - \begin{bmatrix} A_{e_1,i_1}A_{i_1,i_1}^{-1}g_{i_1} \\ A_{e_2,i_2}A_{i_2,i_2}^{-1}g_{i_2} \end{bmatrix}. \tag{16}$$

From Theorem (1) it follows that a classical block Jacobi method applied to (15) recovers the Schwarz iteration with Robin transmission conditions for the case $p = ik$. Instead of iterating this system via block Jacobi we apply a Krylov subspace iteration and demonstrate the performance of this simple algebraic decomposition at the example of the solution of a Helmholtz equation $-\Delta u - k^2 u = 0$ on the unit square $[0,1]^2$. The mesh is a regular triangular mesh with 200 elements. The basis on each mesh consists of 16 equally spaced plane wave directions leading to an overall system size of $n = 3200$. On the boundary of the domain impedance conditions are applied, such that the exact solution is a Hankel source $H_0(k|x-\hat{y}|)$ with

$\hat{y} = (-1, -1)$. The GMRES convergence for the solution of the full system (15) for various wavenumbers k is shown in the left plot of Figure 1. The convergence tolerance is set to 10^{-5}. For the simple algebraic decomposition approach in (16) the results become significantly better. The right plot of Figure 1 shows the results for various wavenumbers for the solution of (16). The subdomain solves were performed with UMFPACK as fast sparse direct solver. The overall system size of (16) is $n = 320$. As expected the results deteriorate for higher wavenumbers, which is due to $p = ik$ only being a good parameter for low-frequency problems.

References

1. A. Alonso-Rodriguez and L. Gerardo-Giorda. New nonoverlapping domain decomposition methods for the harmonic Maxwell system. *SIAM J. Sci. Comput.*, 28(1):102–122, 2006.
2. D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of Discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.
3. I Babuska and J M Melenk. The partition of unity method. *Intern. J. Numer. Methods Eng.*, 40(4):727–758, 1997.
4. A H Barnett and T Betcke. Stability and convergence of the method of fundamental solutions for Helmholtz problems on analytic domains. *J. Comp. Phys.*, 227(7003–7026), 2008.
5. A H Barnett and T Betcke. An exponentially convergent nonpolynomial finite element method for time-harmonic scattering from polygons. *SIAM J. Sci. Comput.*, 32(3):1417–1441, 2010.
6. Olivier Cessenat and Bruno Despres. Application of an Ultra Weak Variational Formulation of Elliptic PDEs to the Two-Dimensional Helmholtz Problem. *SIAM J. Numer. Anal.*, 35(1):255–299, 1998.
7. V. Dolean, L. Gerardo-Giorda, and M. J. Gander. Optimized Schwarz methods for Maxwell equations. *SIAM J. Scient. Comp.*, 31(3):2193–2213, 2009.
8. Stanley C Eisenstat. On the rate of convergence of the Bergman-Vekua method for the numerical solution of elliptic boundary value problems. *SIAM J. Numer. Anal.*, 11:654–680, 1974.
9. O. Ernst and M.J. Gander. Why it is difficult to solve Helmholtz problems with classical iterative methods. In O. Lakkis I. Graham, T. Hou and R. Scheichl, editors, *Numerical Analysis of Multiscale Problems*, pages 325–363. Springer Verlag, 2012.
10. C Farhat, R Tezaur, and J Toivanen. A domain decomposition method for discontinuous Galerkin discretizations of Helmholtz problems with plane waves and Lagrange multipliers. *Intern. J. Numer. Methods Eng.*, 78(13):1513–1531, 2009.
11. M. J. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 24(1):38–60, 2002.
12. Martin J. Gander, Laurence Halpern, and Frédéric Magoulès. An optimized Schwarz method with two-sided robin transmission conditions for the Helmholtz equation. *Int. J. for Num. Meth. in Fluids*, 55(2):163–175, 2007.
13. M.J. Gander. Schwarz methods over the course of time. *ETNA*, 31:228–255, 2008.
14. M.J. Gander and S. Hajian. Block Jacobi for discontinuous Galerkin discretizations: no ordinary Schwarz methods. In *Domain Decomposition Methods in Science and Engineering XXI*, Lect. Notes Comput. Sci. Eng. Springer, same volume, 2014.
15. Claude J Gittelson, Ralf Hiptmair, and Ilaria Perugia. Plane wave discontinuous Galerkin methods: analysis of the h-version. *M2AN Math. Model. Numer. Anal.*, 43(2):297–331, 2009.
16. R Hiptmair, A Moiola, and I Perugia. Plane wave discontinuous Galerkin methods for the 2D Helmholtz equation: analysis of the p-version. *SIAM J. Numer. Anal.*, 49(1):264–284, 2011.
17. A Moiola, R Hiptmair, and I Perugia. Plane wave approximation of homogeneous Helmholtz solutions. *Z. Angew. Math. Phys.*, 62(5):809–837, July 2011.
18. P Monk and Da-Qing Wang. A least-squares method for the Helmholtz equation. *Comput. Methods Appl. Mech. Engrg.*, 175(1-2):121–136, 1999.

Optimized Schwarz Methods for curl-curl time-harmonic Maxwell's equations

Victorita Dolean[1,2], Martin J. Gander[2], Stéphane Lanteri[3], Jin-Fa Lee[4], and Zhen Peng[4]

1 Introduction

Like the Helmholtz equation, the high frequency time-harmonic Maxwell's equations are difficult to solve by classical iterative methods. Domain decomposition methods are currently most promising: following the first provably convergent method in [4], various optimized Schwarz methods were developed over the last decade [2, 3, 10, 11, 1, 6, 13, 14, 16, 8]. There are however two basic formulations for Maxwell's equation: the first order formulation, for which complete optimized results are known [6], and the second order, or curl-curl formulation, with partial optimization results [1, 13, 16]. We show in this paper that the convergence factors and the optimization process for the two formulations are the same. We then show by numerical experiments that the Fourier analysis predicts very well the behavior of the algorithms for a Yee scheme discretization, which corresponds to Nedelec edge elements on a tensor product mesh, in the curl-curl formulation. When using however mixed type Nedelec elements on an irregular tetrahedral mesh, numerical experiments indicate that transverse magnetic (TM) modes are less well resolved for high frequencies than transverse electric (TE) modes, and a heuristic can then be used to compensate for this in the optimization.

2 Optimized Schwarz algorithms

We consider the curl-curl problem in a bounded domain Ω, with boundary conditions on $\partial\Omega$ such that the problem is well posed [12]. A general Schwarz algorithm then solves for $n = 1, 2\ldots$ and the decomposition $\Omega = \Omega_1 \cup \Omega_2$ the subdomain problems

$$\begin{aligned} -\omega^2 \mathbf{E}^{1,n} + \nabla \times \left(\nabla \times \mathbf{E}^{1,n}\right) &= -i\omega Z \mathbf{J} \quad \text{in } \Omega_1 \\ \mathscr{T}_{\mathbf{n}_1}(\mathbf{E}^{1,n}) &= \mathscr{T}_{\mathbf{n}_1}(\mathbf{E}^{2,n-1}) \text{ on } \partial\Omega_1 \cap \Omega_2, \\ -\omega^2 \mathbf{E}^{2,n} + \nabla \times \left(\nabla \times \mathbf{E}^{2,n}\right) &= -i\omega Z \mathbf{J} \quad \text{in } \Omega_2 \\ \mathscr{T}_{\mathbf{n}_2}(\mathbf{E}^{2,n}) &= \mathscr{T}_{\mathbf{n}_2}(\mathbf{E}^{1,n-1}) \text{ on } \partial\Omega_2 \cap \Omega_1, \end{aligned} \tag{1}$$

[1] University of Nice-Sophia Antipolis, France, e-mail: dolean@unice.fr .[2] University of Geneva, Switzerland, e-mail: martin.gander@unige.ch .[3]Inria Sophia Antipolis-Méditerranée, France, e-mail: Stephane.Lanteri@inria.fr .[4] The Ohio State University, ElectroScience Laboratory, e-mail: jinlee@esl.eng.ohio-state.edu; e-mail: peng.98@osu.edu

where $\Gamma_{12} = \partial\Omega_1 \cap \Omega_2$, $\Gamma_{21} = \partial\Omega_2 \cap \Omega_1$, and $\mathscr{T}_{\mathbf{n}_j}$ are transmission conditions. The classical Schwarz method uses for example the impedance condition $\mathscr{T}_{\mathbf{n}}(\mathbf{E}) = (\nabla \times E \times \mathbf{n}) \times \mathbf{n} + i\omega \mathbf{E} \times \mathbf{n}$, where $\mathbf{n}$ denotes the unit outward normal.

The transmission conditions in [6] for the first order formulation, for which complete optimization results are available, can be written for the curl-curl formulation in the form

$$\mathscr{T}_{\mathbf{n}}^{DGG}(\mathbf{E}) = (I + \gamma_1(\mathscr{S}_{TM} + \mathscr{S}_{TE}))(\nabla \times \mathbf{E} \times \mathbf{n}) \times \mathbf{n} + i\omega(I - \gamma_1(\mathscr{S}_{TM} + \mathscr{S}_{TE}))(\mathbf{E} \times \mathbf{n}), \tag{2}$$

where $\mathscr{S}_{TM} = \nabla_\tau \nabla_\tau \cdot$, $\mathscr{S}_{TE} = \nabla_\tau \times \nabla_\tau \times$ and τ denotes the tangential direction. These transmission conditions are a particular case of the more general formulation

$$\begin{aligned} \mathscr{T}_{\mathbf{n}}^{1}(\mathbf{E}) = {} & (I + \nu_1(\delta_1 \mathscr{S}_{TM} + \delta_2 \mathscr{S}_{TE}))(\nabla \times \mathbf{E} \times \mathbf{n}) \times \mathbf{n} \\ & + i\omega(I - \nu_2(\delta_3 \mathscr{S}_{TM} + \delta_4 \mathscr{S}_{TE}))(\mathbf{E} \times \mathbf{n}), \end{aligned} \tag{3}$$

since by choosing $\delta_1 = \delta_2 = \delta_3 = \delta_4 = 1$, $\nu_1 = \nu_2 = \gamma_1$ in (3) we obtain (2).

Rawat and Lee proposed in [16] a transmission condition of the form

$$\begin{aligned} \mathscr{T}_{\mathbf{n}}^{RL}(\mathbf{E}) &= \mathbf{n} \times \nabla \times \mathbf{E} + \alpha \mathbf{n} \times (\mathbf{E} \times \mathbf{n}) + \beta \nabla_\tau \times \nabla_\tau \times (\mathbf{n} \times \mathbf{E} \times \mathbf{n}) + \gamma \nabla_\tau \nabla_\tau \cdot \mathbf{n} \times (\nabla \times \mathbf{E}) \\ &= (I + \gamma \mathscr{S}_{TM})(\mathbf{n} \times \nabla \times \mathbf{E}) + (\alpha + \beta \mathscr{S}_{TE})(\mathbf{n} \times (\mathbf{E} \times \mathbf{n})), \end{aligned} \tag{4}$$

and analyzed the performance for the case of plane waves traveling in the *yz* plane and with the interface in the *xy* plane. A different choice of transmission conditions was proposed in [13],

$$\begin{aligned} \mathscr{T}_{\mathbf{n}}^{TETM}(\mathbf{E}) = {} & (I - \gamma_2(\delta_1 \mathscr{S}_{TM} + \mathscr{S}_{TE}))(\mathbf{n} \times \nabla \times \mathbf{E}) \\ & + i\omega(-I + \gamma_2(\mathscr{S}_{TM} + \delta_4 \mathscr{S}_{TE}))(\mathbf{n} \times (\mathbf{E} \times \mathbf{n})). \end{aligned} \tag{5}$$

Both transmission conditions (4) and (5) are a particular case of the more general formulation

$$\begin{aligned} \mathscr{T}_{\mathbf{n}}^{2}(\mathbf{E}) = {} & (I + \nu_1(\delta_1 \mathscr{S}_{TM} + \delta_2 \mathscr{S}_{TE}))(\mathbf{n} \times \nabla \times \mathbf{E}) \\ & + i\omega(-I + \nu_2(\delta_3 \mathscr{S}_{TM} + \delta_4 \mathscr{S}_{TE}))(\mathbf{n} \times (\mathbf{E} \times \mathbf{n})), \end{aligned} \tag{6}$$

since by taking $\delta_1 = \delta_4 = 1$, $\delta_2 = \delta_3 = 0$, $\nu_1 = \gamma$, $\nu_2 = \beta$ in (6) we obtain (4), and choosing $\delta_2 = \delta_3 = 1$, $\nu_1 = -\gamma_2$, $\nu_2 = \gamma_2$ in (6) we obtain (5).

Thus, at first sight, it seems that there are two different classes of optimized algorithms, the ones with transmission conditions (3), and the ones with (6). One can show however that the optimized algorithm with the special form (2) of the transmission conditions (3) has identical convergence properties to the algorithm with transmission conditions (6) when taking $\delta_1 = \delta_2 = \delta_3 = \delta_4 = 1$, $\nu_1 = \nu_2 = -\gamma_1$ in (6), see [5]. In the following we will thus simply denote $\mathscr{T}_{\mathbf{n}}^2$ by $\mathscr{T}_{\mathbf{n}}$ and only study that case.

3 Convergence analysis using the TE-TM decomposition

We use Fourier analysis, and thus assume that the coefficients are constant, and the domain on which the original problem is posed is $\Omega = \mathbb{R}^3$, in which case we need the Silver-Müller radiation condition $\lim_{r\to\infty} r(\nabla \times E \times \mathbf{n} + i\omega\mathbf{E}) = 0$, where $r = |\mathbf{x}|$, $\mathbf{n} = \mathbf{x}/|\mathbf{x}|$, in order to obtain well-posed problems [12]. The two subdomains are now half spaces, $\Omega_1 = (0,\infty)\times\mathbb{R}^2$, $\Omega_2 = (-\infty, L)\times\mathbb{R}^2$, the interfaces are $\Gamma_{12} = \{L\}\times\mathbb{R}^2$ and $\Gamma_{21} = \{0\}\times\mathbb{R}^2$, and the overlap is $L \geq 0$. Let the Fourier transform in y and z directions be $\mathscr{F}\mathbf{E}(x,y,z) = \int_{\mathbb{R}^2} \mathbf{E}(x,y,z)e^{i(k_y y + k_z z)}dydz$, where we denote by k_y and k_z the Fourier variables and $|\mathbf{k}|^2 = k_y^2 + k_z^2$. We first compute the local solutions of the homogeneous counterparts of (1), which corresponds to the equation that the error satisfies at each iteration.

Lemma 1 (Local solutions). *The local solutions of (1) with* $\mathbf{J} = 0$*, computed in Fourier space, are given by*

$$\mathscr{F}(\mathbf{E}^1) = e^{\lambda x}\left(-\frac{i(A_2k_z + A_4k_y)}{\lambda}, A_4, A_2\right)^T, \quad \mathscr{F}(\mathbf{E}^2) = e^{-\lambda x}\left(\frac{i(A_1k_z + A_3k_y)}{\lambda}, A_3, A_1\right)^T \tag{7}$$

where $\lambda = \sqrt{|\mathbf{k}|^2 - \omega^2}$ *and the coefficients* $A_{1,2,3,4}$ *may depend on* k_y*,* k_z*.*

The expressions of the solutions in Lemma 1 suggest a different formulation in another basis, which we call the TE-TM decomposition. It can easily be obtained by splitting the solution in (7) into combinations of solutions verifying $A_2k_z + A_4k_y = 0, A_2, A_4 \neq 0$ (TE modes) and $A_2k_y = A_4k_z, A_2, A_4 \neq 0$ (TM modes).

Lemma 2 (Local solution decomposition into TE-TM modes). *The local solutions in (7) can be re-written as*

$$\mathscr{F}(\mathbf{E}^j) = A_{TM}\mathscr{F}(\mathbf{E}^{j,TM}) + A_{TE}\mathscr{F}(\mathbf{E}^{j,TE}), \quad j = 1,2, \tag{8}$$

where

$$\begin{aligned}
\mathscr{F}(\mathbf{E}^{1,TE}) &= e^{\lambda x}\left(0, -\tfrac{k_z}{k_y}, 1\right)^T, \mathscr{F}(\mathbf{E}^{1,TM}) = e^{\lambda x}\left(-\tfrac{i|\mathbf{k}|^2}{k_y\lambda}, 1, \tfrac{k_z}{k_y}\right)^T, \\
\mathscr{F}(\mathbf{E}^{2,TE}) &= e^{-\lambda x}\left(0, -\tfrac{k_z}{k_y}, 1\right)^T, \mathscr{F}(\mathbf{E}^{2,TM}) = e^{-\lambda x}\left(\tfrac{i|\mathbf{k}|^2}{k_y\lambda}, 1, \tfrac{k_z}{k_y}\right)^T.
\end{aligned} \tag{9}$$

To derive the convergence factors, we compute the action of the interface operators from (6), and then replace them into the interface iterations of (1). This calculation is greatly simplified with the decomposition into TE-TM modes, with the difference that we now iterate on the unknowns A_{TE} and A_{TM}. The convergence factor is again given by the spectral radius of some iteration matrix, as in [6], and this matrix happens to be conveniently diagonal for a certain choice of the parameters.

Theorem 1 (Convergence factor for the TE-TM decomposition). *In the case* $\delta_3 = \delta_2$*,* $\delta_4 = \frac{1}{\delta_1}$*, which holds for all algorithms we consider, the interface iteration can be written as*

$$\begin{bmatrix} A_{TE} \\ A_{TM} \end{bmatrix}^{1,n} = B \begin{bmatrix} A_{TE} \\ A_{TM} \end{bmatrix}^{1,n-2},$$

with the interface iteration matrix B given by

$$B = \frac{\lambda - i\omega}{\lambda + i\omega} \begin{bmatrix} -\frac{(\lambda+i\omega)(\lambda\nu_2\delta_2+i\omega\nu_1\delta_1)+1}{(\lambda-i\omega)(-\lambda\nu_2\delta_2+i\omega\nu_1\delta_1)-1} & 0 \\ 0 & \frac{(\lambda+i\omega)(\lambda\nu_1\delta_1\delta_2+i\omega\nu_2)+\delta_1}{(\lambda-i\omega)(-\lambda\nu_1\delta_1\delta_2+i\omega\nu_2)-\delta_1} \end{bmatrix} e^{-2\lambda L}. \quad (10)$$

The proof can be found in [5]. The convergence factor of the algorithm is for each Fourier mode given by the spectral radius of B. In the following we assume that there is no overlap, $L = 0$.

Corollary 1 (DGG conditions). *If we choose $\delta_1 = 1$, $\delta_2 = 1$, $\nu_1 = \nu_2 = -\frac{1}{|\mathbf{k}|^2 - 2\omega^2 + 2i\omega s}$ in (10), where s is a complex parameter to be chosen, we obtain an iteration matrix with the same convergence factor as in the first order formulation in [6],*

$$\rho_{DGG}(|\mathbf{k}|, \omega, s) = \left| \frac{\sqrt{|\mathbf{k}|^2 - \omega^2} - i\omega}{\sqrt{|\mathbf{k}|^2 - \omega^2} + i\omega} \cdot \frac{\sqrt{|\mathbf{k}|^2 - \omega^2} - s}{\sqrt{|\mathbf{k}|^2 - \omega^2} + s} \right|. \quad (11)$$

Corollary 2 (RL conditions). *If we choose $\delta_1 = 1$, $\delta_2 = 0$, $\nu_1 = \frac{1}{\omega^2 + \omega\tilde{k}^{tm}}$, $\nu_2 = \frac{1}{\omega^2 + \omega\tilde{k}^{te}}$ in (10), where $\tilde{k}^{tm}$ and $\tilde{k}^{te}$ are real parameters to be chosen, we obtain an iteration matrix with convergence factor as in [16],*

$$\rho_{RL}(|\mathbf{k}|, \omega, \tilde{k}^{te}, \tilde{k}^{tm}) = \left| \frac{\sqrt{|\mathbf{k}|^2 - \omega^2} - i\omega}{\sqrt{|\mathbf{k}|^2 - \omega^2} + i\omega} \right| \cdot \max\left(\left| \frac{\sqrt{|\mathbf{k}|^2 - \omega^2} - i\tilde{k}^{te}}{\sqrt{|\mathbf{k}|^2 - \omega^2} + i\tilde{k}^{te}} \right|, \left| \frac{\sqrt{|\mathbf{k}|^2 - \omega^2} - i\tilde{k}^{tm}}{\sqrt{|\mathbf{k}|^2 - \omega^2} + i\tilde{k}^{tm}} \right| \right). \quad (12)$$

Corollary 3 (TETM conditions). *If we choose $\delta_1 = \frac{i\omega + s^{te}}{i\omega + s^{tm}}$, $\delta_2 = 1$, $\nu_1 = \nu_2 = -\frac{1}{|\mathbf{k}|^2 - 2\omega^2 + i\omega(s^{te} + s^{tm})}$ in (10), where s^{tm} and s^{te} are real parameters to be chosen, we obtain an iteration matrix with convergence factor as in [14],*

$$\rho_{TETM}(|\mathbf{k}|, \omega, s^{tm}, s^{te}) = \left| \frac{\sqrt{|\mathbf{k}|^2 - \omega^2} - i\omega}{\sqrt{|\mathbf{k}|^2 - \omega^2} + i\omega} \right| \cdot \max\left\{ \left| \frac{\sqrt{|\mathbf{k}|^2 - \omega^2} - s^{te}}{\sqrt{|\mathbf{k}|^2 - \omega^2} + s^{te}} \right|, \left| \frac{\sqrt{|\mathbf{k}|^2 - \omega^2} - s^{tm}}{\sqrt{|\mathbf{k}|^2 - \omega^2} + s^{tm}} \right| \right\}. \quad (13)$$

It remains to explain the choice of the parameters in the three different algorithms: for the DGG conditions, the same choice as for the first order formulation can be used. Minimizing the maximum over all relevant frequencies leads for example in [6, case 3, section 3.5] to

$$s = (1+i)\sqrt{k^{max}}(k_+^2 - \omega^2)^{1/4}/\sqrt{2}, k^{max} = \frac{C}{h} \quad (14)$$

with k_+ an estimate of the closest numerical frequency just above ω.

For the RL conditions, the authors in [16, 13] recommend

$$\tilde{k}^{te} = -i\sqrt{\left(\tfrac{1}{2}(k^{max,te} + \omega)\right)^2 - \omega^2}, \quad \tilde{k}^{tm} = -i\sqrt{\left(\tfrac{1}{2}(k^{max,tm} + \omega)\right)^2 - \omega^2}, \quad (15)$$

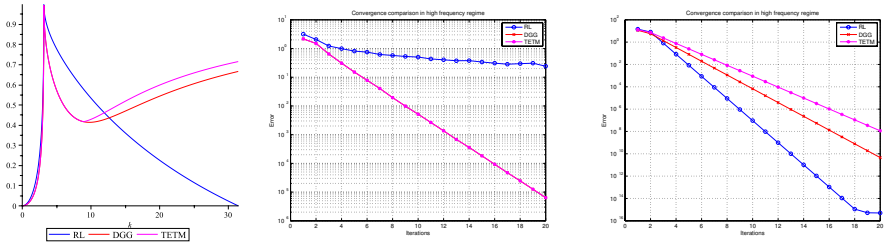


Fig. 1 Comparison of the theoretical contraction factors (11), (12), and (13) on the left, and convergence histories of the corresponding algorithms, in the middle with a random initial guess, and on the right with a high frequency initial guess

with the same estimates $k^{max,te}$, $k^{max,tm}$ as in the TETM case, where a separate minimization of the maximum leads to the parameters

$$s^{te} = (1+i)\sqrt{k^{max,te}}(k_+^2 - \omega^2)^{1/4}/\sqrt{2}, s^{tm} = (1+i)\sqrt{k^{max,tm}}(k_+^2 - \omega^2)^{1/4}/\sqrt{2}. \tag{16}$$

For a mixed type Nedelec elements on irregular tetrahedral meshes, numerical observations in [15, Section 4.5.1] indicate that a good choice is $k^{max,te} = k^{max}$, $k^{max,tm} = \frac{2}{3}k^{max}$. If however $k^{max,te} = k^{max,tm}$, as it is for example the case in a Yee discretization, then minimizing the maximum of the contraction factor in TETM leads again to the DGG transmission conditions. Note that a separate optimization for the TE and TM modes can also potentially be beneficial if one knows for example a priori which TE or TM modes one wants to simulate, since one can then optimize the performance of the algorithm for these modes.

4 Numerical results

We first show a comparison of the theoretical convergence factors ρ_{RL}, ρ_{DGG} and ρ_{TETM} in Figure 1 on the left for the specific values $h = 0.001$ and $\omega = 10\pi$. From these convergence factors, we can expect that a numerical implementation of the algorithm with all error frequencies contained in the initial guess will overall converge better with the DGG and TETM conditions than with the RL conditions. The DGG and TETM transmission conditions have identical convergence behavior for lower error frequencies, but for high error frequencies, the DGG conditions are better. Even though being much less favorable in general, the RL conditions are excellent for very high frequency evanescent error modes.

We now illustrate our convergence results with numerical experiments. We first solve Maxwell's equations in the curl-curl formulation on the domain $\Omega = (0,\pi)^2 \times (0,2\pi)$ using a Yee scheme. We decompose the domain into two subdomains $\Omega_1 = (0,\pi)^2 \times (0,\pi)$ and $\Omega_2 = (0,\pi)^2 \times (\pi,2\pi)$. We chose the frequency $\omega = 1$ for this experiment. We show in Figure 1 in the middle and on the right the convergence

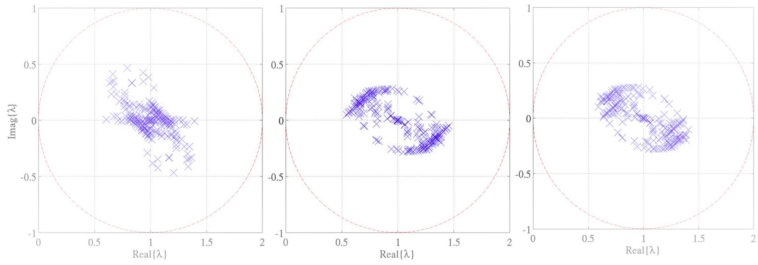


Fig. 2 Eigenspectra for a parallel plate waveguide, $h = \lambda_0/4$, $p = 2$, RL (left), DGG (middle), TETM (right)

histories for the three Schwarz algorithms we considered over 20 iterations. In the middle, we used a random initial guess to make sure all frequencies are present in the error. Here the DGG and TETM algorithms have identical convergence behavior, while the RL algorithm is very slow as expected from the theoretical result in the left plot. On the right we used the highest possible frequency that can be represented on the mesh only as the initial guess for the error. Now, the RL conditions lead to the fastest convergence, whereas the TETM conditions are the slowest, again as expected from the theoretical plot on the left. This shows that one has to be careful when doing numerical investigations: from the right panel in Figure 1, one could conclude that the RL conditions are the best, but this only holds for one particular error frequency. This is why one solves min-max problems to determine optimized parameters: the algorithm needs to be good for all error frequencies uniformly, see especially the experiments in [9, Section 5.1, Figure 5.2].

Next, we show numerical experiments for a discretization with mixed type Nedelec elements on irregular tetrahedral grids. We start by examining the eigenvalues of three non-overlapping domain decomposition matrices, using the RL, DGG, and TETM conditions. We chose a $0.5\lambda_0$ (λ_0 denotes the free space wavelength) segment of a parallel plate waveguide with both ports terminated by first order absorbing boundary conditions. The parallel plate waveguide is partitioned by a transverse plane into two equally sized sub-domains. The mesh size is chosen to be $\lambda_0/4$. In Figure 2, we show the eigenvalue distributions of the three iteration matrices using the RL, DGG, and TETM transmission conditions. All of them provide desirable convergence properties, since all the eigenvalues are within the shifted-unit-circle. It is clear that the spectral radius of the DGG conditions is slightly smaller than the RL conditions, due to the fact that $\rho_{\mathrm{DGG}}^{\max} < \rho_{\mathrm{RL}}^{\max}$. We also see that for this discretization, the TETM conditions further improve the convergence factor of the TM modes: one portion of eigenvalues moves towards the center of the unit circle.

We now present scalability studies: we denote by d the size of the sub-domains, by D the size of the entire problem domain and by h the mesh size. A Krylov subspace iterative method, Generalized Conjugate Residual (GCR) [7], is used for the solution of the matrix equation.

Scalability with respect to ωh: we simulate a $1.5\lambda_0$ segment of a parallel plate waveguide. The waveguide is partitioned into three sub-domains, each $0.5\lambda_0$ long.

Table 1 Number of iterations to attain a relative residual reduction of 10^{-8} for different transmission conditions and different mesh sizes

Cases	$\omega h = 1.57$	$\omega h = 0.785$	$\omega h = 0.524$	$\omega h = 0.393$
RL conditions	23 (19)	27 (17)	34 (22)	41 (22)
DGG conditions	21 (18)	26 (21)	32 (19)	39 (20)
TETM conditions	21 (14)	25 (15)	30 (12)	36 (14)

These sub-domains are meshed independently and quasi-uniformly such that the interface meshes do not match. The mesh size varies from $h = \lambda_0/4$ to $h = \lambda_0/16$. The numbers of iterations required using the RL, DGG, and TETM transmission conditions are given in Table 1, for a random initial guess, and in parentheses with the TEM mode as an excitation and a zero initial guess. The $h-$refinement permits the representation of more high frequency evanescent modes on the interface, and we see that computing just one TEM mode solution with a zero initial guess requires much less iterations than when all modes are present. The iteration numbers could still substantially be lowered in the one TEM mode case by optimizing just for that mode.

Scalability with respect to ωD: We fix the subdomain size to $0.3\lambda_0$, and we increase the length of the waveguide by increasing the number of subdomains. The mesh size is kept fixed as well at $h = \lambda_0/8$. The performance of the methods for 10, 20, 40, and 80 subdomains is shown in Table 2, again for a random initial guess, and then in parentheses with the TEM mode as excitation, and a zero initial guess. In this study, the propagating modes are of pre-dominant significance since the wave must travel from one end of the waveguide to the other. We see that all of the three conditions show dependence on the problem size, which is expected in the absence of a coarse space. We see that the DGG and TETM conditions perform much better in this set of experiments than the RL condition, and also that all methods need a substantially bigger number of iterations in the presence of all error modes, than when just one mode is present.

5 Conclusions

We have shown that the optimized transmission conditions developed for the first order Maxwell system in [6] can also be used for the curl-curl formulation, and

Table 2 Number of iterations to attain a relative residual reduction of 10^{-8} for different transmission conditions and different problem sizes

Cases	$\omega D = 18.8$	$\omega D = 37.7$	$\omega D = 75.3$	$\omega D = 150.7$
RL conditions	34 (17)	63 (28)	146 (72)	363 (168)
DGG conditions	30 (18)	49 (22)	90 (33)	185 (51)
TETM conditions	31 (21)	46 (22)	85 (29)	176 (37)

the corresponding convergence factors and hence optimized parameters are identical. We illustrated these results with a Yee discretization of the curl-curl formulation. We then showed also numerical experiments with a mixed type Nedelec finite element discretization on irregular tetrahedral grids, and presented several scaling experiments.

References

1. Alonso-Rodriguez, A., Gerardo-Giorda, L.: New nonoverlapping domain decomposition methods for the harmonic Maxwell system. SIAM J. Sci. Comput. **28**(1), 102–122 (2006)
2. Chevalier, P., Nataf, F.: An OO2 (Optimized Order 2) method for the Helmholtz and Maxwell equations. In: 10th International Conference on Domain Decomposition Methods in Science and in Engineering, pp. 400–407. AMS, Boulder, Colorado, USA (1997)
3. Collino, P., Delbue, G., Joly, P., Piacentini, A.: A new interface condition in the non-overlapping domain decomposition for the Maxwell equations. Comput. Methods Appl. Mech. Engrg. **148**, 195–207 (1997)
4. Després, B., Joly, P., Roberts, J.: A domain decomposition method for the harmonic Maxwell equations. In: Iterative methods in linear algebra, pp. 475–484. North-Holland, Amsterdam (1992)
5. Dolean, V., Gander, M.J., Lee, J.F., Peng, Z.: Optimized Schwarz methods for solving the curl-curl time-harmonic Maxwell equations. submitted (2013)
6. Dolean, V., Gerardo-Giorda, L., Gander, M.J.: Optimized Schwarz methods for Maxwell equations. SIAM J. Scient. Comp. **31**(3), 2193–2213 (2009)
7. Eisenstat, S., Elman, H., Schultz, M.: Variational iterative methods for nonsymmetric systems of linear equations. SIAM Journal on Numerical Analysis **20**(2), 345–357 (1983)
8. El Bouajaji, M., Dolean, V., Gander, M.J., Lanteri, S.: Optimized Schwarz methods for the time-harmonic Maxwell equations with dampimg. SIAM J. Scient. Comp. **34**(4), 2048–2071 (2012)
9. Gander, M.J.: Schwarz methods over the course of time. Electronic Transactions on Numerical Analysis **31**, 228–255 (2008)
10. Gander, M.J., Magoulès, F., Nataf, F.: Optimized Schwarz methods without overlap for the Helmholtz equation. SIAM J. Sci. Comput. **24**(1), 38–60 (2002)
11. Lee, S.C., Vouvakis, M., Lee, J.F.: A non-overlapping domain decomposition method with non-matching grids for modeling large finite antenna arrays. J. Comput. Phys. **203**(1), 1–21 (2005)
12. Nedelec, J.C.: Acoustic and electromagnetic equations. Integral representations for harmonic problems. Applied Mathematical Sciences, 144. Springer Verlag (2001)
13. Peng, Z., Lee, J.F.: Non-conformal domain decomposition method with second-order transmission conditions for time-harmonic electromagnetics. J. Comput. Phys. **229**(16), 5615–5629 (2010)
14. Peng, Z., Rawat, V., Lee, J.F.: One way domain decomposition method with second order transmission conditions for solving electromagnetic wave problems. J. Comput. Phys. **229**(4), 1181–1197 (2010)
15. Rawat, V.: Finite element domain decomposition with second order transmission conditions for time-harmonic electromagnetic problems. Ph.D. thesis, Ohio State University (2009)
16. Rawat, V., Lee, J.F.: Nonoverlapping domain decomposition with second order transmission condition for the time-harmonic Maxwell's equations. SIAM J. Sci. Comput. **32**(6), 3584–3603 (2010)

On the Origins of Iterative Substructuring Methods

Martin J. Gander[1] and Xuemin Tu[2]

1 The Invention of Substructuring Methods

Substructuring methods were invented in the engineering community. A very early precursor is the so called "Moment Distribution Method", or "Hardy Cross Method" named after its inventor [11]. Cross states in the introduction to his paper from 1930 his motivation for the method:

> The reactions in beams, bents, and arches which are immovably fixed at their ends have been extensively discussed. They can be found comparatively readily by methods which are more or less standard. The method of analysis herein presented enables one to derive from these the moments, shears, and thrusts required in the design of complicated continuous frames.

The idea is to give a precise method how to combine structures for which their reaction to load is known (i.e. tabulated), when they interact at joints between structures. The method is iterative, and described in Figure 1.

In modern terms, it is a Jacobi relaxation applied to the displacement formulation of structural analysis [39], but also a precursor to the finite element method.

> The method of moment distribution is this: (*a*) Imagine all joints in the structure held so that they cannot rotate and compute the moments at the ends of the members for this condition; (*b*) at each joint distribute the unbalanced fixed-end moment among the connecting members in proportion to the constant for each member defined as "stiffness"; (*c*) multiply the moment distributed to each member at a joint by the carry-over factor at that end of the member and set this product at the other end of the member; (*d*) distribute these moments just "carried over"; (*e*) repeat the process until the moments to be carried over are small enough to be neglected; and (*f*) add all moments —fixed-end moments, distributed moments, moments carried over—at each end of each member to obtain the true moment at the end.
>
> To the mathematically inclined the method will appear as one of solving a series of normal simultaneous equations by successive approximation. From an engineering viewpoint it seems simpler and more useful to think of the solution as if it were a physical occurrence.

Fig. 1 The Hardy Cross Method from 1930

[1] Section de Mathématiques, Université de Genève, CP 64, 1211 Genève, Switzerland e-mail: Martin.Gander@unige.ch [2] Department of Mathematics, University of Kansas, 1460 Jayhawk Blvd, Lawrence, KS 66045-7594, U.S.A. e-mail: xtu@math.ku.edu

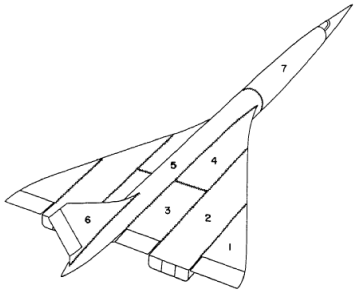


FIG. 3. Typical substructure arrangement for delta aircraft.

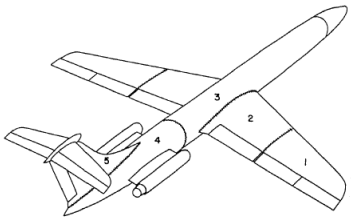


FIG. 4. Typical substructure arrangement for conventional aircraft.

Fig. 2 Two plane structures with non-overlapping subdomain decompositions from the original publication of Przemieniecki in 1963

It was however at Boeing, right after the reinvention of the finite element method for the design of aircraft [38, 9], where Przemieniecki introduced in his seminal paper [33] the first substructuring method of the form we know them now. He first explains why substructuring became necessary:

> The necessity for dividing a structure into substructures arises either from the requirement that different types of analysis have to be used on different components, or because the capacity of the digital computer is not adequate to cope with the analysis of the complete structure.

At the time, computational techniques for the simulation of aircraft were rapidly developing, and complex structures had to be simulated, as shown in the original drawings of Przemieniecki in Figure 2. Unlike in the case of Cross, the substructures were too complicated to have tabulated solutions, and had to be simulated as well. At the beginning of his paper, Przemieniecki describes the idea of his domain domain decomposition method, which is not so different from the method of Cross, but it is not iterative:

> In the present method each substructure is first analyzed separately, assuming that all common boundaries with adjacent substructures are completely fixed: these boundaries are then relaxed simultaneously and the actual boundary displacements are determined from the equations of equilibrium of forces at the boundary joints. The substructures are then analyzed separately again under the action of specified external loading and the previously determined boundary displacements.

Let us see how this can be written in mathematical terms, using the notation used by Przemieniecki. Like for many structural engineers at that time, the reasoning was at the discrete level: let P be the exterior forces, K the stiffness matrix, and U the displacement vector. Then these quantities satisfy the system of equations

$$KU = P. \tag{1}$$

We now partition the unknowns U into unknowns U_i in the interior of each substructure, and the unknowns U_b on the boundaries between substructures, as indicated in

Figure 2. If we partition the matrix and right hand side accordingly, the system (1) can be rewritten as

$$\begin{bmatrix} K_{bb} & K_{bi} \\ K_{ib} & K_{ii} \end{bmatrix} \begin{bmatrix} U_b \\ U_i \end{bmatrix} = \begin{bmatrix} P_b \\ P_i \end{bmatrix}. \tag{2}$$

Now the algorithm of Przemieniecki has three steps, as we have seen above. The first step must keep boundaries between substructures fixed, and hence an (unknown) force $P^{(\alpha)}$ is needed to keep these boundaries fixed. Przemieniecki therefore partitions the forcing vector into

$$P = P^{(\alpha)} + P^{(\beta)} = \begin{bmatrix} P_b^{(\alpha)} \\ P_i \end{bmatrix} + \begin{bmatrix} P_b^{(\beta)} \\ 0 \end{bmatrix}. \tag{3}$$

Since with the first vector on the right hand side as a load, the boundaries of the substructures do not move, the displacements can also be written in the same decomposition, namely

$$U = U^{(\alpha)} + U^{(\beta)} = \begin{bmatrix} 0 \\ U_i^{(\alpha)} \end{bmatrix} + \begin{bmatrix} U_b \\ U_i^{(\beta)} \end{bmatrix}. \tag{4}$$

By linearity, we can rewrite the original system as two systems, which represent the first two steps in Przemieniecki's algorithm,

$$(\alpha): \begin{bmatrix} K_{bb} & K_{bi} \\ K_{ib} & K_{ii} \end{bmatrix} \begin{bmatrix} 0 \\ U_i^{(\alpha)} \end{bmatrix} = \begin{bmatrix} P_b^{(\alpha)} \\ P_i \end{bmatrix},$$

and

$$(\beta): \begin{bmatrix} K_{bb} & K_{bi} \\ K_{ib} & K_{ii} \end{bmatrix} \begin{bmatrix} U_b \\ U_i^{(\beta)} \end{bmatrix} = \begin{bmatrix} P_b^{(\beta)} \\ 0 \end{bmatrix}.$$

In the first step of Przemieniecki's algorithm one needs to solve the first system. Because the interfaces between substructures are not moving, this system simplifies to

$$K_{bi} U_i^{(\alpha)} = P_b^{(\alpha)}, \qquad K_{ii} U_i^{(\alpha)} = P_i.$$

Knowing the forces P_i in the interior of each substructure, we can compute the interior displacements when the interfaces are kept fixed, $U_i^{(\alpha)} = K_{ii}^{-1} P_i$. Inserting this result into the first equation uncovers the unknown force that Przemieniecki needed to impose to keep the interfaces fixed, namely

$$P_b^{(\alpha)} = K_{bi} K_{ii}^{-1} P_i.$$

We can now determine the remaining forces $P_b^{(\beta)}$ on the interfaces,

$$P_b^{(\beta)} = P_b - P_b^{(\alpha)} = P_b - K_{bi} K_{ii}^{-1} P_i,$$

and inserting this result into the second system (β) gives

$$K_{bb}U_b + K_{bi}U_i^{(\beta)} = P_b^{(\beta)}, \qquad K_{ib}U_b + K_{ii}U_i^{(\beta)} = 0.$$

We can now compute the second step in Przemieniecki's algorithm, namely the response of the structures to the interface loading $P_b^{(\beta)}$. The second equation gives the internal displacement $U_i^{(\beta)}$ based on the boundary displacement U_b,

$$U_i^{(\beta)} = -K_{ii}^{-1}K_{ib}U_b,$$

and inserting this into the first equation, Przemieniecki obtains for the unknowns at the interfaces the system

$$(K_{bb} - K_{bi}K_{ii}^{-1}K_{ib})U_b = P_b - K_{bi}K_{ii}^{-1}P_i. \tag{5}$$

We see that the procedure, which Przemieniecki motivated by a strictly mechanical argument, leads simply to the Schur complement system, where all interior variables are eliminated! We note that the Schur complement system can also be derived using discrete harmonic functions on the substructures. The third and last step, after solving the Schur complement system, is to simply compute the corresponding interior displacements, and the problem is solved. Historically, the Schur complement was also known under the name capacitance matrix [25], as we will see next.

2 Capacitance Matrix Methods

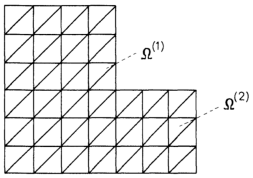
The capacitance matrix method became popular in the early 1970, due to a publication by Buzbee, Dorr, George and Golub [8] that has a very short abstract:

> There are several very fast direct methods which can be used to solve the discrete Poisson equation on rectangular domains. We show that these methods can also be used to treat problems on irregular regions.

The paper first gives a general introduction to Schur complement techniques at the algebraic level, and then the authors show how Schur complements can be used to solve problems on irregular domains by imbedding, and by domain splitting, with a typical example of an L-shaped domain. As in Przemieniecki, the Schur complement system (5) is solved by direct methods. A new important idea was then introduced by Proskurowski and Widlund in [32]:

> This new formulation leads to well-conditioned capacitance matrix equations which can be solved quite efficiently by the conjugate gradient method. A highly accurate solution can, therefore, be obtained at an expense which grows no faster than that for a fast Laplace solver on a rectangle when the mesh size is decreased.

The authors explain that their method can use fast Possion solvers for a similar purpose as for the fundamental solustions when constructing the classical integral equations of potential theory. The key contribution is however the solution of the Schur complement system by a Krylov method, which paved the way for iterative substructuring methods.



$$\gamma_0 (K^{1/2} y, y)_{R^p} \leqq (C y, y)_{R^p} \leqq \gamma_1 (K^{1/2} y, y)_{R^p}$$

γ_0 *and* γ_1 *are the following constants*

$$\gamma_0 = r/(1+(1+r^2)^{1/2}), \qquad \gamma_1 = (1+2r(1+2\sigma))\, b_2/(4r\, b_1)$$

$$r = (32)^{1/2}(a_2 - a_1)/b_1 .$$

Fig. 3 Original Figure by Dryja [16] to introduce preconditioned iterative substructuring methods on the left, and first preconditioner estimate on the right

3 Iterative Substructuring Methods

The explicit calculation of the Schur complement S is expensive and requires large amount of memory since the matrix is much denser than the original stiffness matrix K as defined in (1), even though it is much smaller. However the action of the Schur complement on a vector can be calculated implicitly by solving local substructure problems. Therefore the explicit formation of the Schur complement can be avoided if Krylov space methods are used to solve the interface problem (5) iteratively, as shown in [32]. To make the number of iterations however manageable, for certain accuracy, it is crucial to construct a suitable preconditioner for the Krylov subspace methods. In a sequence of papers [15, 16, 17], Dryja first introduced preconditioned Krylov space methods for solving the interface problem (5). The L-shaped domain shown in Figure 3 on the left was divided into two subdomains in [16], and the preconditioner is selected as $K^{-1/2}$, where K is here the discrete Laplacian operator on the subdomain interface. Dryja proved in [16] the first spectral equivalence result for preconditioning the capacitance matrix, as shown in Figure 3 on the right. This result was proved using Fourier analysis, and the preconditioner can also be implemented efficiently using a fast sine transform.

Golub and Mayers proposed a slightly improved version of this preconditioners in [24]. In [1, 2], Bjørstad and Widlund explicit derived and diagonalized the local Schur complement $S^{(i)}$ and proposed two preconditioners. The preconditioner considered by Dryja was called the "good method" and the other, the Neumann-Dirichlet preconditioner for two subdomains, the "excellent method". The application of this preconditioner to a vector requires the solution of one subdomain Neumann problem and one subdomain Dirichlet problem. It converges in one step if the two subdomains come from a symmetric region cut in half and the triangulation is regular and symmetric. If the subdomain partition allows a red-black coloring, the Neumann-Dirichlet (Dirichlet-Neumann) algorithms can also be extended to many subdomains.

Another type of preconditioner, the Neumann-Neumann preconditioner was introduced in [22, 3, 27]. The application of this preconditioner to a vector requires the solutions of two Dirichlet problems and two Neumann problems. Thus, it is more expensive than the Neumann-Dirichlet preconditioner. However, it is easy to extend to many subdomains and can be made to perform well with jump coefficients by introducing a simple scaling operator.

The number of iterations will increase with an increase of the number of subdomains for most one-level preconditioners. An additional level is needed to remove such dependence. For Dirichlet-Neumann preconditioners, when the subdomain partitions has cross points, a natural second, coarse level solver can be formed using variables related to these cross points, see [19, 18]. The two-level Neumann-Neumann algorithms, known as Balancing Neumann-Neumann algorithms, where introduced in [29, 28, 26, 21]. The coarse level solver can be constructed using weighted counting functions. The balancing Neumann-Neumann algorithm has been extended to several applications such as for the mixed finite element discretizations, Stokes, and almost incompressible elasticity, [10, 31, 23]. Recently, the balancing domain decomposition by constraints method has been developed and it has been widely used [12, 30]; it is similar to the balancing Neumann-Neumann algorithms but its coarse problems are given in terms of a set of primal constraints partially enforcing contiunity across the interface.

4 Primal Iterative Substructuring Methods

There is another class of substructuring methods known as the primal iterative substructuring methods. The difference between the preconditioners in this class and the algorithms described in Section 3 is that the coupling between all pairs of faces, edges, and vertices are eliminated in the preconditioners of this class while the coupling between neighboring subdomains are eliminated in the previous class.

The development of the primal iterative substructuring methods started with a famous series of four papers [4, 5, 6, 7]. [4] is the first paper on iterative substructuring methods to deal with cross points satisfactorily. The algorithm proposed in that paper has a coarse level component formed in terms of the cross points and an almost optimal condition number bound was established in two dimensions. However such a coarse level problem does not always work well in three dimensions because of a much weaker finite element Sobolev inequality. Related methods with a coarse solver based on the wire basket were introduced in [7] for three dimensional problems and an almost optimal bound was obtained.

An observation on using a change of basis from a partial hierarchical basis to the usual nodal basis for this class of algorithms was made in [37] and many preconditioners of this type were introduced in [20]. These algorithms have been successfully implemented and extended to three dimensional linear elasticity [35, 36]. Quite recently, the coarse components introduced here have also been used for overlapping domain decomposition methods to obtain algorithms independent of the coefficient jumps [34, 13, 14].

Acknowledgements This work was supported in part by National Science Foundation Contract No. DMS-1115759.

References

1. Petter E. Bjørstad and Olof B. Widlund. Solving elliptic problems on regions partitioned into substructures. In Garrett Birkhoff and Arthur Schoenstadt, editors, *Elliptic Problem Solvers II*, pages 245–256, New York, 1984. Academic Press.
2. Petter E. Bjørstad and Olof B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.*, 23(6):1093–1120, 1986.
3. Jean-François Bourgat, Roland Glowinski, Patrick Le Tallec, and Marina Vidrascu. Variational formulation and algorithm for trace operator in domain decomposition calculations. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods. Second International Symposium on Domain Decomposition Methods*, pages 3–16, Philadelphia, PA, 1989. SIAM. Los Angeles, California, January 14–16, 1988.
4. James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, I. *Math. Comp.*, 47(175):103–134, 1986.
5. James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, II. *Math. Comp.*, 49(179):1–16, 1987.
6. James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, III. *Math. Comp.*, 51(184):415–430, 1988.
7. James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, IV. *Math. Comp.*, 53(187):1–24, 1989.
8. B.L Buzbee, F.W. Dorr, J.A. George, and G.H. Golub. The direct solution of the discrete Poisson equation on irregular regions. *SIAM J. Numer. Anal.*, 8(4):722–736, 1971.
9. Ray W. Clough. The finite element method in plane stress analysis. In *Proc ASCE Conf Electron Computat, Pittsburg, PA*, 1960.
10. Lawrence C. Cowsar, Jan Mandel, and Mary F. Wheeler. Balancing domain decomposition for mixed finite elements. *Math. Comp.*, 64(211):989–1015, July 1995.
11. Hardy Cross. Analysis of continuous frames by distributing fixed-end moments. *Transactions of the American Society of Civil Engineers*, (Paper 1793):1–10, 1930.
12. Clark R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, 25(1):246–258, 2003.
13. Clark R. Dohrmann and Olof B. Widlund. An overlapping Schwarz algorithm for almost incompressible elasticity. *SIAM J. Numer. Anal.*, 47(4):2897–2923, 2009.
14. Clark R. Dohrmann and Olof B. Widlund. Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity. *Internat. J. Numer. Methods Engrg.*, 82(2):157–183, 2010.
15. Maksymilian Dryja. An algorithm with a capacitance matrix for a variational-difference scheme. In Guri I. Marchuk, editor, *Variational-Difference Methods in Mathematical Physics*, pages 63–73, Novosibirsk, 1981. USSR Academy of Sciences.
16. Maksymilian Dryja. A capacitance matrix method for Dirichlet problem on polygon region. *Numer. Math.*, 39:51–64, 1982.
17. Maksymilian Dryja. A finite element-capacitance method for elliptic problems on regions partitioned into subregions. *Numer. Math.*, 44:153–168, 1984.
18. Maksymilian Dryja. A method of domain decomposition for 3-D finite element problems. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 43–61, Philadelphia, PA, 1988. SIAM. Paris, France, January 7–9, 1987.
19. Maksymilian Dryja, Wlodek Proskurowski, and Olof Widlund. A method of domain decomposition with crosspoints for elliptic finite element problems. In Blagovest Sendov, editor, *Optimal Algorithms*, pages 97–111, Sofia, Bulgaria, 1986. Bulgarian Academy of Sciences.
20. Maksymilian Dryja, Barry F. Smith, and Olof B. Widlund. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM J. Numer. Anal.*, 31(6):1662–1694, December 1994.
21. Maksymilian Dryja and Olof B. Widlund. Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. *Comm. Pure Appl. Math.*, 48(2):121–155, February 1995.

22. Roland Glowinski and Mary F. Wheeler. Domain decomposition and mixed finite element methods for elliptic problems. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 144–172, Philadelphia, PA, 1988. SIAM. Paris, France, January 7–9, 1987.
23. Paulo Goldfeld, Luca F. Pavarino, and Olof B. Widlund. Balancing Neumann-Neumann preconditioners for mixed approximations of heterogeneous problems in linear elasticity. *Numer. Math.*, 95(2):283–324, 2003.
24. Gene Golub and D. Mayers. The use of preconditioning over irregular regions. In R. Glowinski and Jacques-Louis. Lions, editors, *Computing Methods in Applied Sciences and Engineering, VI*, pages 3–14, Amsterdam, New York, Oxford, 1984. North-Holland. Proceedings of a conference held in Versailles, France, December 12-16,1983.
25. R.W. Hockney. Potential calculation and some application. *Methods Comput. Phys.*, 9:135–211, 1970.
26. Patrick Le Tallec. Domain decomposition methods in computational mechanics. In J. Tinsley Oden, editor, *Computational Mechanics Advances*, volume 1 (2), pages 121–220. North-Holland, 1994.
27. Patrick Le Tallec, Yann-Hervé De Roeck, and Marina Vidrascu. Domain decomposition methods for large linearly elliptic three-dimensional problems. *J. Comput. Appl. Math.*, 34(1):93–117, 1991.
28. Jan Mandel. Balancing domain decomposition. *Comm. Numer. Meth. Engrg.*, 9:233–241, 1993.
29. Jan Mandel and Marian Brezina. Balancing domain decomposition: Theory and computations in two and three dimensions. Technical Report UCD/CCM 2, Center for Computational Mathematics, University of Colorado at Denver, 1993.
30. Jan Mandel and Clark R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Linear Algebra Appl.*, 10(7):639–659, 2003.
31. Luca F. Pavarino and Olof B. Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. *Comm. Pure Appl. Math.*, 55(3):302–335, March 2002.
32. W. Proskurowski and O. Widlund. On the numerical solution of helmholtz's equation by the capacitance matrix method. *Math. of Comp.*, 30(135):433–468, 1976.
33. J. S. Przemieniecki. Matrix structural analysis of substructures. *Am. Inst. Aero. Astro. J.*, 1:138–147, 1963.
34. Marcus V. Sarkis. Nonstandard coarse spaces and Schwarz methods for elliptic problems with discontinuous coefficients using non-conforming elements. *Numer. Math.*, 77(3):383–406, 1997.
35. Barry F. Smith. *Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity*. PhD thesis, Courant Institute of Mathematical Sciences, September 1990. Tech. Rep. 517, Department of Computer Science, Courant Institute.
36. Barry F. Smith. An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems. *SIAM J. Sci. Statist. Comput.*, 13(1):364–378, January 1992.
37. Barry F. Smith and Olof B. Widlund. A domain decomposition algorithm using a hierarchical basis. *SIAM J. Sci. Stat. Comput.*, 11(6):1212–1220, 1990.
38. N. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp. Stiffness and deflection analysis of complex structures. *J. Aero. Sci.*, 23:805–23, 1956.
39. K. Y. Volokh. On foundation of the Hardy Cross method. *International Journal of Solids and Structures*, 39(16):4197–4200, 2002.

Discontinuous Coarse Spaces for DD-Methods with Discontinuous Iterates

Martin J. Gander[1], Laurence Halpern[2], and Kévin Santugini Repiquet[3]

1 Introduction

Basic iterative domain decomposition methods (DDM) can only transmit information between direct neighbors. Such methods never converge in less iterations than the diameter of the connectivity graph between subdomains. Convergence rates are dependent on the number of subdomains, and thus algorithms are not scalable. The use of a coarse space [16] is the only way to provide information from distant subdomains, as they enable global information transfer, ensuring scalability. In this respect, well known methods are the two level additive Schwarz method [3], and the FETI [13] and balancing Neumann-Neumann methods [12, 4, 14]. See also [11] for non-symmetric problems. For complete analyses of such scalable methods, see [18, 17].

Adding an effective coarse space correction to an existing method is currently an active area of research, for example in the case of high contrast problems [2, 15]. Combining coarse spaces with methods with discontinuous iterates, such as optimized Schwarz methods (OSM [8]) is also non-trivial, see [6] and chapter 5 in [5] which contain extensive numerical tests, and [7] for a rigorous analysis of a special case. For restricted Additive Schwarz (RAS [1]), which also produces discontinuous global iterates since they are glued from local ones by the $\widetilde{R}$ operators in RAS in an aribirary fashion, see [9] in the present proceedings. We explain in §2 why an effective coarse space for non-overlapping OSM (and DDMs with discontinuous iterates in general) should inherently be discontinuous. In §3, we present one possible realization of a coarse grid correction based on a discontinuous coarse space, and we show that convergence in one coarse correction step can be obtained, although this is only practical in one dimension. For higher dimensional problems, we then propose approximations of this optimal coarse space. In §4, we present numerical experiments with this new algorithm, and finally give an outlook on future work in §5.

[1] Université de Genève, Section de mathmatiques, e-mail: Martin.Gander@unige.ch [2] Laboratoire Analyse, Géométrie & Applications UMR 7539 CNRS, Université PARIS 13, 93430 VILLETANEUSE, FRANCE, e-mail: halpern@math.univ-paris13.fr [3] Université Bordeaux, IMB, CNRS UMR5251, MC2, INRIA Bordeaux - Sud-Ouest e-mail: Kevin.Santugini@math.u-bordeaux1.fr

2 Choosing a good coarse space

In this section, we explain why it makes sense to consider discontinuous coarse space corrections. We place ourselves in a continuous setting and consider the model problem

$$\eta u - \Delta u = f \quad \text{in } \Omega, \qquad \gamma u = 0 \quad \text{on } \partial\Omega, \tag{1}$$

where Ω is a polygonal domain in $\mathbb{R}^d$ ($d \geq 1$), and γ is a trace operator.

Let $(\Omega_i)_{1\leq i\leq N}$ be a non-overlapping domain decomposition of Ω. A non-overlapping optimized Schwarz method with a coarse grid correction is given in Algorithm 1.

Algorithm 1 (Generic)

Initialize u_i^0, either by zero or using the coarse solution.
for $n \geq 0$ and until convergence **do**
  In each subdomain Ω_i, compute the uncorrected iterates $u_i^{n+1/2}$ in parallel using the optimized Schwarz algorithm.
  Compute a coarse correction U^{n+1} belonging to a coarse space X.
  Set the corrected iterates to $u_i^{n+1} := u_i^{n+1/2} + U^{n+1}$.
end for
Set either $u_i := u_i^{n-1/2}$ or $u_i := u_i^n$ where n is the exit index of the above loop.

Instead of explaining in detail how the coarse correction U^{n+1} is computed, we first focus on the more important question of how to choose the coarse space X.

2.1 Suboptimality of a conformal coarse space

We first explain why with a coarse space $X \subset H^1(\Omega)$, it is not possible to compute a very good coarse correction for a domain decomposition method with discontinuous iterates. A function u, with $u_{|\Omega_i}$ in $H^1(\Omega)$, is a weak solution of (1) if

(i) u satisfies (1) inside each subdomain Ω_i,
(ii) u has no jump between two adjacent subdomains,
(iii) the normal derivative of u has no jump between two adjacent subdomains.

In an efficient domain decomposition algorithm, each step of the algorithm should improve as many of these three conditions as possible. In particular, the coarse grid correction should be such that the iterates u_i^{n+1} are closer to satisfying these three conditions than the uncorrected iterates $u_i^{n+1/2}$. However:

(i) The uncorrected iterates already satisfy the equation inside each subdomain.
(ii) The uncorrected iterates are discontinuous, they have jumps in the Dirichlet traces along interfaces.
(iii) The uncorrected iterates have also discontinuous normal derivatives, they have jumps in the Neumann traces along interfaces.

Using continuous coarse functions is suboptimal for a method that produces discontinuous iterates, since they can not reduce the Dirichlet jumps. Using instead a discontinuous coarse space, for example P_0, then the Dirichlet jumps can be improved, but not the Neumann jumps. If the coarse functions are even more regular, for example $\mathscr{C}^1$ on the whole domain, then neither the Dirichlet jumps nor the Neumann jumps can be improved.

2.2 Better coarse spaces for methods with discontinuous iterates

To be effective, a coarse space for a domain decomposition method that produces discontinuous iterates must contain discontinuous functions. Furthermore the discontinuities must be aligned with the interfaces between subdomains. Suppose that the subdomains Ω_i form a conforming polygonal mesh $\mathscr{T}^{\Omega}$ of Ω (triangles or rectangles in two dimensions). The local polynomial space $\mathscr{P}_1$ is $\mathbb{P}_1$ in the former case, $\mathbb{Q}_1$ in the latter. The conforming coarse space is $\mathscr{P}_1(\mathscr{T}^{\Omega}) = \{v \in \mathscr{C}^0(\bar{\Omega}),\ \forall i,\ v_{|\Omega_i} \in \mathscr{P}_1\}$, but a better choice is the discontinuous coarse space (or broken in the Discontinuous Galerkin literature) $\mathscr{P}_1^{\text{disc}}(\mathscr{T}^{\Omega}) = \{v,\ \forall i,\ v_{|\Omega_i} \in \mathscr{P}_1\}$, where the continuity across the interfaces is no longer required.

In addition, for linear problems, it is important for the coarse shape functions to be solutions of the homogeneous counterpart of equation (1) inside each subdomain, because then the corrected iterates are also solutions of the interior equation inside each subdomain. To see this, it suffices to note that the error between the monodomain solution and any iterates produced by the optimized Schwarz method is always a solution to the homogeneous equation inside each subdomain. Therefore, in $H_0^{1,disc}(\Omega) = \{u \in L^2(\Omega),\ \forall i,\ u_{|\Omega_i} \in H^1(\Omega_i),\ u = 0 \text{ on } \partial\Omega\}$, the space

$$\mathscr{A} = \{u \in H_0^{1,disc}(\Omega),\ \forall i,\ (\eta - \triangle)u_{|\Omega_i} = 0\} \tag{2}$$

is an ideal candidate for a coarse space. For one-dimensional problems, the space $\mathscr{A}$ is finite dimensional, and can directly be used as the coarse space. In higher dimensions, the space $\mathscr{A}$ is infinite dimensional for the continuous problem, and must therefore be discretized as well to be practical: a finite dimensional subspace of $\mathscr{A}$ must be chosen. To do so, one only needs to choose boundary conditions on each $\partial\Omega_i$. For the particular algorithm presented in the next section, the intersection of the coarse space with H^1 should be "big enough", for there to be enough test functions. This can be guaranteed by constructing coarse elements with potentially compatible Dirichlet conditions.

For these reasons we introduce the space of all discontinuous functions, whose element shape functions are solutions to the homogeneous equation inside each subdomain, with Dirichlet boundary conditions:

$$\mathscr{P}_1^{\mathscr{A}}(\mathscr{T}^{\Omega}) = \{u \in \mathscr{A},\ \exists \hat{u} \in \mathscr{P}_1^{\text{disc}}(\mathscr{T}^{\Omega}),\ u = \hat{u} \text{ in } \bigcup_{i=1}^{N} \partial\Omega_i\}. \tag{3}$$

Remark 1. Other Dirichlet boundary conditions can be used to define the discontinuous coarse elements. Any finite dimensional vector space of continuous functions defined over $\bigcup_{i=1}^{N} \partial\Omega_i$ may be used to construct finite dimensional coarse spaces that are subsets of $\mathscr{A}$ with a "big enough" continuous subset.

Now that we have chosen the coarse space, we can design an efficient algorithm to compute a discontinuous coarse space correction. The coarse correction must be chosen such that it diminishes both Dirichlet and Neumann jumps while not losing too much in terms of satisfying the interior equations in each subdomain. Using the full coarse space $\mathscr{A}$ (which is only practical in one dimension), any good algorithm for computing the coarse correction should converge in a single coarse iteration, because the error between the iterates and the exact solution belongs to $\mathscr{A}$. In the next section, we present such an algorithm, the DCS-DMNV algorithm (discontinuous coarse space - Dirichlet minimizer Neumann variational), which is suitable for finite element methods.

3 The DCS-DMNV algorithm

We formulate the algorithm with subdomain iterates at the continuous level, with a discrete coarse space.

Let X_d be any finite dimensional coarse space, subspace of $H_0^{1,disc}(\Omega)$ (for example $\mathscr{P}_1^{\mathscr{A}}(\mathscr{T}^{\Omega})$ defined above), and $X_c = X_d \cap H^1(\Omega)$, which will be non-trivial if we use potentially compatible Dirichlet boundary conditions for the coarse elements. We define the positive quadratic form over $H_0^{1,disc}(\Omega)$ by

$$q : H_0^{1,disc}(\Omega) \to \mathbb{R}^+, \quad u \mapsto \sum_{ij} \int_{\partial\Omega_i \cap \partial\Omega_j} \left|u_i - u_j\right|^2 \mathrm{d}\sigma.$$

Note that the kernel of q is $H^1(\Omega)$. The DCS-DMNV algorithm is stated in Algorithm 2 at page 515.

Proposition 1 (Existence of the coarse iterate). *Let $(u_i^{n+1/2})_{1\le i\le N}$ be the local iterates. Then there exists a unique U^{n+1} in X_d that satisfies* (5)*.*

Proof. The function $V \mapsto q(u^{n+1/2}+V)$ is quadratic, choose one minimizer U_d^{n+1} on X_d. By Lax-Milgram's Lemma, there exists a unique U_c^{n+1} in X_c such that $U^{n+1} = U_c^{n+1} + U_d^{n+1}$ satisfies (5b). Uniqueness comes from the fact that q is quadratic. □

The DCS-DMNV algorithm 2 has the important property of converging in a single coarse step if the full coarse space $\mathscr{A}$ is used. However, it is only practical in a one dimensional setting as the coarse space is too big in higher dimensions. We state that theorem in the discrete case.

Theorem 1 (Convergence in a single coarse step for the full coarse space). *Let Ω be a bounded polygonal domain in $\mathbb{R}^d$. Let $(\Omega_i)_{1\le i\le N}$ be a domain decomposition*

Algorithm 2 (DCS-DMNV)

Initialize u_i^0 by either zero or $u^0_{|\Omega_i}$ where u^0 is the coarse solution.
while no convergence **do**
Compute the local iterates $u_i^{n+1/2} \in H^1(\Omega_i)$ in parallel by

$$\eta u_i^{n+1/2} - \triangle u_i^{n+1/2} = f \quad \text{in } \Omega_i, \tag{4a}$$

$$\frac{\partial u_i^{n+1/2}}{\partial n_i} + p u_i^{n+1/2} = \frac{\partial u_j^n}{\partial n_i} + p u_j^n \quad \text{on } \partial\Omega_i \cap \Omega_j, \tag{4b}$$

$$u_i^{n+1/2} = 0 \quad \text{on } \partial\Omega_i \cap \Omega. \tag{4c}$$

Define a global $u^{n+1/2} \in H_0^{1,disc}(\Omega)$ as $u_i^{n+1/2}$ in Ω_i. Set U^{n+1} as the unique function in X_d such that

$$q(u^{n+1/2} + U^{n+1}) = \min_{v \in X_d} q(u^{n+1/2} + v), \tag{5a}$$

and satisfying

$$\eta \int_\Omega U^{n+1}(x)v(x)\mathrm{d}x + \int_\Omega \nabla U^{n+1}(x)\nabla v(x)\mathrm{d}x = -\sum_{i,j} \int_{\partial\Omega_i \cap \partial\Omega_j} \left(\frac{\partial u_i^{n+1/2}}{\partial n_i} + \frac{\partial u_j^{n+1/2}}{\partial n_j} \right) v \mathrm{d}\sigma, \tag{5b}$$

for all test functions v in X_c.
Set $u_i^{n+1} := u_i^{n+1/2} + U^{n+1}$.
end while
Set $u := u_i^{n-1/2}$ on Ω_i for each i in $\{1, \ldots, N\}$.

of Ω that also forms a coarse mesh of Ω. Let $\mathscr{T}_h$ be a simplicial or a cartesian fine mesh on Ω which is a refinement of the $(\Omega_i)_{1\le i\le N}$ domain decomposition. Let $\mathscr{F}$ be the conformal finite element space given either by $P_1(\mathscr{T}_h)$ if $\mathscr{T}_h$ is simplicial or by $Q_1(\mathscr{T}_h)$ if $\mathscr{T}_h$ is cartesian. Let $\mathscr{F}^{\mathrm{disc}}$ be the set of all functions on Ω whose restriction to each Ω_i is also the restriction of a function belonging to $\mathscr{F}$ to Ω_i, $\mathscr{F}_0$ be the space of functions in $\mathscr{F}$ vanishing on all subdomain boundaries.

Let $X_d \subset \mathscr{F}^{\mathrm{disc}}$ be a coarse space. Suppose all elements in X_d satisfy the homogenous variational equation for all test functions in $\mathscr{F}_0$. Let $X_c = X_d \cap \mathscr{F}$. Suppose $u \mapsto ((u(x_{i,j}))_{1\le j\le k_i})_{1\le i\le N}$ is from X_d onto $\prod_{i=1}^N \mathbb{R}^{k_i}$ where k_i is the number of nodes of $\mathscr{T}_h$ located on $\partial\Omega_i \setminus \partial\Omega$ and where $x_{i,j}$ is the j-th node located on $\partial\Omega_i \setminus \partial\Omega$. Then, for any choice of initial fine iterate $(u_i^0)_{1\le i\le N}$ satisfying the variational equation for all test functions in $\mathscr{F}_0$, the DCS-DMNV algorithm 2 converges in a single coarse step.

Proof. Let $(U_i)_{1\le N}$ be the coarse correction. Let $u_i^1 = u_i^0 + U_i$ be the corrected iterates. The corrected iterates must satisfy the minimum jump condition (5a). Since $u \mapsto ((u(x_{i,j}))_{1\le j\le k_i})_{1\le i\le N}$ is onto, it is possible to completely cancel the jumps, therefore $q((u_i^1)_{1\le i\le N}) = 0$ and u^1 defined over Ω as $u^1_{|\Omega_i} = u_i^1$ belongs to $\mathscr{F}$, *i.e.* is continuous across subdomains. Moreover, since the coarse correction satisfies the

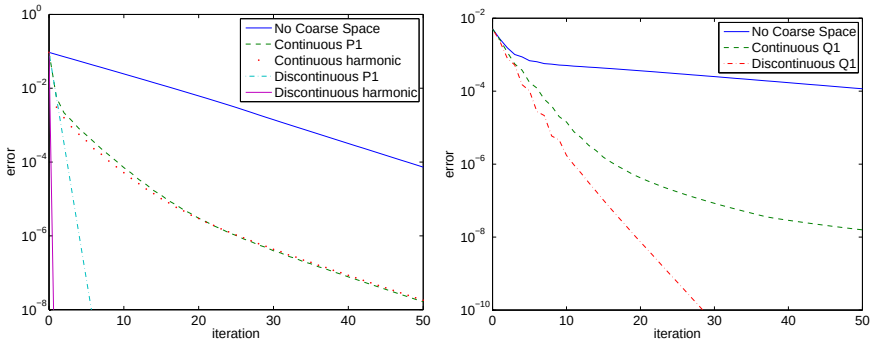


Fig. 1 Convergence curves of the DCS-DMNV algorithm in 1*D* (left) and 2*D* (right)

homogenous counterpart of (1) inside each subdomain, the corrected iterates satisfy the variational equation for all test functions in $\mathscr{F}_0$. By (5b), the corrected iterates also satisfy the variational equation for all test functions v in X_c. Since $u \mapsto ((u(x_{i,j}))_{1\le j\le k_i})_{1\le i\le N}$ is onto, $\mathscr{F} = X_c + \mathscr{F}_0$. Therefore, the corrected iterates satisfy the variational equation for all test functions in $\mathscr{F}$. □

In a practical implementation however, convergence in a single coarse iteration would only be possible if the coarse space contains all the degrees of freedom on the interfaces corresponding to the fine discretization of the subdomain problems, which would be a very rich and expensive coarse space. We will see in the next section that a linear approximation of all the degrees of freedom on the interfaces already leads to a very good discontinuous coarse correction.

4 Numerical results

We implemented the DCS-DMNV algorithm 2 in one and two dimensions, using a finite element discretization based on a regular cartesian grid. In 1*D*, we chose $\Omega =]0,4[$, $\eta = 10$ and the right hand-side $f(x) = -1$. For the Robin parameter, we used $p = 5$, with 60 subdomains. Convergence curves are presented in Figure 1.

As expected, a coarse grid correction with conforming $\mathbb{P}_1$ finite elements already improves convergence. Requiring the coarse shape functions to be solutions of the homogeneous equation within each subdomain does not bring any further gain. A striking improvement is the use of discontinuous $\mathbb{P}_1$ elements. Optimal convergence (see Theorem 1) can then be reached if in addition the coarse functions solve the homogeneous equation inside each subdomain.

In two-dimension, we chose $\eta = 0$ and iterate directly on the error equations, *i.e.*, we solve $-\Delta e = 0$ but start with random boundary conditions on each subdomain. Q_1 elements discretize $\Omega =]0,4[^2$, and the algorithm is run with $p = 12.4$, 5×5 subdomains and 10×10 cells per subdomain. It is important for the Robin

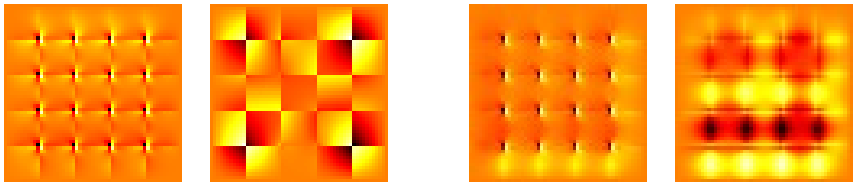


Fig. 2 Error of the algorithm with continuous (left) and discontinuous (right) coarse grid correction at iterations 5 and 20. Each error has been renormalized independently.

boundary conditions to be lumped, see [10]. To compute the coarse correction, we use the Conjugate Gradient algorithm to compute the result of the multiplication of the pseudo-inverse of Q, $q(u,v) = (Qu|v)$ with a right hand-side derived from the uncorrected iterates. This gives us one minimizer in X_d of the q functionnal. To satisfy the variational condition, an additional continuous coarse correction can then be computed in X_c.

As in the $1D$ case, the convergence curves presented in Figure 1 show that the discontinuous coarse space correction leads to a much faster convergence than the continuous one. Even though the discontinuous coarse space is only a subset of the optimal theoretical coarse space, the improvement over continuous coarse spaces is substantial. In order to see in the error how the jumps slow down the convergence of the continuous coarse correction version, we present in Figure 2 a few snapshots of the errors. We observe the formation of a checkerboard like structure which cannot be corrected by a continuous coarse space. Once the errors look like a checkerboard, the convergence of the continuous coarse correction algorithm slows down considerably. Using a discontinuous coarse space prevents the checkerboard like structure from appearing.

5 Conclusion

We have shown that for domain decomposition methods with discontinuous iterates, the use of a discontinuous coarse space greatly improves that of a standard continuous one. We have designed one such discontinuous coarse space algorithm, the DCS-DMNV algorithm, the formulation of which is well suited for finite element discretizations. In practice, this algorithm should be used in conjunction with Krylov acceleration. We intend to study the behavior of the Krylov accelerated DCS-DMNV in a forthcoming paper. We are currently studying such algorithms also for finite difference and finite volumes schemes, and investigating how the optimization parameter p in the transmission conditions interacts with the Dirichlet boundary conditions used in the definition of the coarse space.

References

1. Cai, X.C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM Journal on Scientific Computing **21**, 239–247 (1999)
2. Dolean, V., Nataf, F., Scheichl, R., Spillane, N.: Analysis of a two-level schwarz method with coarse spaces based on local dirichlet to neumann maps. Computational Methods in Applied Mathematics **12**(4), 391–414 (2012). DOI 10.2478/cmam-2012-0027
3. Dryja, M., Widlund, O.B.: An additive variant of the Schwarz alternating method for the case of many subregions. Tech. Rep. 339, also Ultracomputer Note 131, Department of Computer Science, Courant Institute (1987)
4. Dryja, M., Widlund, O.B.: Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. Comm. Pure Appl. Math. **48**(2), 121–155 (1995)
5. Dubois, O.: Optimized Schwarz methods for the advection-diffusion equation and for problems with discontinuous coefficients. Ph.D. thesis, McGill University (2007)
6. Dubois, O., Gander, M.J.: Convergence behavior of a two-level optimized Schwarz preconditioner. In: Domain Decomposition Methods in Science and Engineering XXI. Springer LNCSE (2009)
7. Dubois, O., Gander, M.J., Loisel, S., St-Cyr, A., Szyld, D.: The optimized Schwarz method with a coarse grid correction. SIAM J. on Sci. Comp. **34**(1), A421–A458 (2012)
8. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**(2), 699–731 (2006)
9. Gander, M.J., Halpern, L., Santugini, K.: A new coarse grid correction for RAS. In: Domain Decomposition Methods in Science and Engineering XXI. Springer LNCSE (2014). Same volume.
10. Gander, M.J., Hubert, F., Krell, S.: Optimized Schwarz algorithm in the framework of DDFV schemes. In: Domain Decomposition Methods in Science and Engineering XXI. Springer LNCSE (2014). Same volume.
11. Japhet, C., Nataf, F., Roux, F.X.: Extension of a coarse grid preconditioner to non-symmetric problems. In: J. Mandel, C. Farhat, X.C. Cai (eds.) Domain decomposition methods, 10 (Boulder, CO, 1997), *Contemporary Mathematics*, vol. 218, pp. 279–286. Amer. Math. Soc., Providence, RI (1998)
12. Mandel, J.: Balancing domain decomposition. Communications in Numerical Methods in Engineering **9**(3), 233–241 (1993). DOI 10.1002/cnm.1640090307
13. Mandel, J., Brezina, M.: Balancing domain decomposition for problems with large jumps in coefficients. Math. Comp. **65**, 1387–1401 (1996)
14. Mandel, J., Tezaur, R.: Convergence of a substructuring method with Lagrange multipliers. Numer. Math. **73**, 473–487 (1996)
15. Nataf, F., Xiang, H., Dolean, V., Spillane, N.: A coarse sparse construction based on local Dirichlet-to-Neumann maps. SIAM J. Sci. Comput. **33**(4), 1623–1642 (2011)
16. Nicolaides, R.A.: Deflation conjugate gradients with application to boundary value problems. SIAM J. Num. An, **24**(2), 355–365 (1987). DOI doi:10.1137/0724027
17. Smith, B.F., Bjørstad, P.E., Gropp, W.: Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press (1996)
18. Toselli, A., Widlund, O.: Domain Decomposition Methods - Algorithms and Theory, *Springer Series in Computational Mathematics*, vol. 34. Springer (2004)

A two-level preconditioning framework based on a Richardson iterative process

Thomas Dufaud[1]

1 Introduction

We focus on the solution of a general linear system $Au = f$ by a Krylov type iterative method, where $A \in \mathbb{R}^{m\times m}$ is non-singular, u, $f \in \mathbb{R}^m$. The major drawback of the GCR (Generalized Conjugate Residual) [7] and the GMRES (General Minimum Residual) [8] methods is their convergence rate that depends on the conditioning number $\kappa(A) = \|A\| \, \|A^{-1}\|$.

The convergence rate of these techniques decreases while κ increases and the use of such methods needs preconditioning. In the following we consider left preconditioning. The goal is to solve $M^{-1}Au = M^{-1}f$ with M^{-1} such that $\kappa(M^{-1}A) \ll \kappa(A)$.

Preconditioning can be enhanced by multilevel techniques. Multilevel techniques are known to be robust for scalar elliptic Partial Differential Equations with standard discretization and to enhance the scalability of domain decomposition method such as Restricted Additive Schwarz preconditioning techniques. An issue is their application to linear system encountered in industrial applications which can be derived from non-elliptic PDEs. Moreover, the building of coarse levels algebraically becomes an issue since the only known information is contained in the operator to inverse.

One can consider a coarse space as a space to represent an approximated solution of a smaller dimension than the leading dimension of the system. It is possible to build a coarse level based on a coarse representation of the solution. Drawing our inspiration from the Aitken-SVD methodology [9] dedicated to Schwarz methods, we proposed to construct an approximation space by computing the Singular Value Decomposition of a set of iterated solutions of the Richardson process associated to a given preconditioner.

From a preconditioner M^{-1} associated to a Richardson process:

$$u^k = u^{k-1} + \alpha M^{-1}\left(f - Au^{k-1}\right) \text{ with } \alpha \in \mathbb{R} \tag{1}$$

We propose to build a two-level additive preconditioner M_{2L}^{-1}:

$$M_{2L}^{-1} = M^{-1} + M_c^{-1} \tag{2}$$

where for a basis $\mathbb{U}_q \in \mathbb{R}^{m\times q}$, $M_c^{-1} = \mathbb{U}_q(\mathbb{U}_q^T A \mathbb{U}_q)^{-1}\mathbb{U}_q^T$.

[1] INRIA Rennes - Bretagne Atlantique, Campus universitaire de Beaulieu, 35042 Rennes Cedex FRANCE, e-mail: thomas.dufaud@inria.fr

The plan of the paper is the following. Section 2 describes the methodology to compute an algebraic coarse level from successive iterations of a Richardson process. Numerical investigations with the RAS preconditioner built for real non-symmetric indefinite operator, are performed in section 3. Section 4 concludes the study.

2 Methodology

The idea is to compute a coarse representation of the solution. In [9] a fully algebraic computation of a coarse space is proposed to perform an Aitken acceleration of vectorial sequence generated with an iterative domain decomposition method. In [6] Aitken-SVD Schwarz algorithms were derived for the Aitken Restricted Additive Schwarz preconditioning technique [5].

The choice of constructing the coarse space with the SVD is based on the following properties. Let $G \in \mathbb{R}^{m\times l}$ Assume that the values $\sigma_k, 1 \leq k \leq l$ are ordered in decreasing order and there exists a q such that $\sigma_q > 0$ while $\sigma_q + 1 = 0$. Then G can be decomposed in a dyadic decomposition:

$$G = \sigma_1 U_1 V_1^* + \sigma_2 U_2 V_2^* + \ldots + \sigma_q U_q V_q^*. \tag{3}$$

This means that SVD provides a way to find optimal lower dimensional approximations of a given series of data. More precisely, it produces an orthonormal base for representing the data series in a certain least squares optimal sense. This can be summarized by the theorem of Schmidt-Eckart-Young-Mirsky:

Theorem 1. *A non-unique minimizer X_* of the problem $\min_{X,rankX=q} \|G - X\|_2 = \sigma_{q+1}(G)$, provided that $\sigma_q > \sigma_{q+1}$, is obtained by truncating the dyadic decomposition of 3 to contain the first q terms: $X_* = \sigma_1 U_1 V_1^* + \sigma_2 U_2 V_2^* + \ldots + \sigma_q U_q V_q^*$*

Moreover, the SVD of a matrix is well-conditioned with respect to perturbations of its entries. Consider the matrix $G, B \in \mathbb{R}^{m\times l}$, the Fan inequalities write $\sigma_{q+s+1}(G + B) \leq \sigma_{q+1}(G) + \sigma_{s+1}(B)$ with $q, s \geq 0, q + s + 1 \leq l$. Considering the perturbation matrix E such that $||E|| = O(\varepsilon)$, then $|\sigma_k(G + E) - \sigma_k(G)| \leq \sigma_1(E) = \|E\|_2, \forall k$. This property does not hold for eigenvalues decomposition where small perturbations in the matrix entries can cause a large change in the eigenvalues.

These properties allow us to search an approximation of the solution in the base linked to the SVD of a sequence of vectors obtained by iterating a linearly convergent iterative process.

Here, we propose a general framework which enables to compute algebraically a two-level additive preconditioner from any preconditioner that can be used in a Richardson iterative process. Algorithm 1 shows the steps to compute M_{2L}^{-1} that way. In step 1, we compute the SVD of l successive iterations stored in a matrix

$G \in \mathbb{R}^{m\times l}$ of a Richardson process (1) having a linear convergence, *i.e.* we compute a dyadic decomposition of G, as $G = \mathbb{U}_l \Sigma_l \mathbb{V}_l^T$, with $\mathbb{U}_l \in \mathbb{R}^{m\times l}$ $\Sigma_l \in \mathbb{R}^{l\times l}$ and $\mathbb{V}_l \in \mathbb{R}^{m\times l}$. In step 2, $\mathbb{U}_q$ is made of the first q columns of $\mathbb{U}_l$ with respect to the decreasing of the singular values $\Sigma_{i,i}$, such that $\mathbb{U}_q$ is full rank. This selection is done according to Theorem 1 where $X_q \in \mathbb{R}^{m\times q}$ is a non-unique minimizer of the problem $\min_{X,rankX=q} \|G-X\|_2 = \sigma_{q+1}(G)$, such that $X_q = \mathbb{U}_q \Sigma_q \mathbb{V}_q^T$ and $rk(\mathbb{U}_q) = q$, with $\mathbb{U}_q \in \mathbb{R}^{m\times q}$, $\Sigma_q \in \mathbb{R}^{q\times q}$ and $\mathbb{V}_q \in \mathbb{R}^{m\times q}$. Once this basis of the coarse space is defined, one can compute the coarse operator (step 3) and solve the coarse problem (step 4).

Algorithm 1 Computation of M_{2L}^{-1} with SVD of solutions of a Richardson process

Require: $(u^k)_{0\le k\le l-1}$, l successive iterates satisfaying $u^{k+1}-u^\infty = \left(I-\alpha M^{-1}A\right)\left(u^k-u^\infty\right)$ starting from any initial guess u^0

1: Compute the Singular Value Decomposition of the snapshots $G = \left[u^0,\ldots,u^{l-1}\right] = \mathbb{U}_l \Sigma_l \mathbb{V}_l^T$
2: Set the index q such that $q = \max_{0\le i\le l-1}\{\Sigma(i,i) > tol\}$, to define the full rank matrix $\mathbb{U}_q = [U_0, U_1, \ldots, U_q]$ {ex.: $tol = 10^{-12}$.}
3: Define the coarse operator $A_c \in \mathbb{R}^{q\times q}$ such that $A_c = \mathbb{U}_q^T A \mathbb{U}_q$
4: Define the two-level additive preconditioner $M_{2L}^{-1} = M^{-1} + \mathbb{U}_q A_c^{-1} \mathbb{U}_q^T$

It is possible to see this approach as a way to approximate a Krylov subspace. Basically, the solution of the linear system $Au = f$ defined in Section 1 consists on minimizing $F(u^k) = \left(f-Au^k, f-Au^k\right)$ on a Krylov space $K_l\left(A, r^0\right) = \{r^0, Ar^0, \ldots, A^{l-1}r^0\} = \{d^0,\ldots,d^{l-1}\}$, where from an arbitrary initial solution $u^0 \in \mathbb{R}^m$, $r^0 = f - Au^0$.

Let choose $u^0 = 0$. Each iterate u^k of the Richardson process can be written in a Krylov subspace:

$$u^k = \sum_{i=0}^{k} \beta_i \left(M^{-1}A\right)^i M^{-1} f\,, \quad \beta_i \neq 0$$

Following Algorithm 1, we can write that

$$span\left(U_0,\ldots,U_{q-1}\right) \subset span\left(U_0,\ldots,U_{l-1}\right)$$

Then the solution of the coarse problem is an approximation of a solution in $span(K_l(M^{-1}A, M^{-1}f))$.

This link enable us to choose a good initial guess for the Krylov method preconditioned by this two-level preconditioning approach by computing the solution $u_c \in \mathbb{R}^q$ of the coarse linear system: $A_c u_c = f_c$, with $f_c = \mathbb{U}_q f$.

Then we can set the initial guess for the Krylov method such that,

$$u^0 = \mathbb{U}_q u_c$$

3 Numerical experiments

In this section we propose to apply the methodology for a RAS preconditioner for the solution of CFD problems. The considered matrices A are real, non-symmetric, indefinite and possibly not positive.

The Additive Schwarz (AS) preconditioning is built from the adjacency graph $G = (W,E)$ of A, where $W = \{1,2,...,m\}$ and $E = \{(i,j) : a_{ij} \neq 0\}$ are the edges and vertices of G. Starting with a non-overlapping partition $W = \cup_{i=1}^{p} W_{i,0}$ and $\delta \geq 0$ given, the overlapping partition $\{W_{i,\delta}\}$ is obtained defining p partitions $W_{i,\delta} \supset W_{i,\delta-1}$ by including all the immediate neighbouring vertices of the vertices in the partition $W_{i,\delta-1}$. Then the restriction operator $R_{i,\delta} : W \to W_{i,\delta}$ defines the local operator $A_{i,\delta} = R_{i,\delta} A R_{i,\delta}^T, A_{i,\delta} \in \mathbb{R}^{m_{i,\delta} \times m_{i,\delta}}$ on $W_{i,\delta}$. The AS preconditioning writes: $M_{AS,\delta}^{-1} = \sum_{i=1}^{p} R_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta}$. Introducing $\tilde{R}_{i,\delta}$ the restriction matrix on a non-overlapping subdomain $W_{i,0}$, the Restricted Additive Schwarz (RAS) iterative process [2] writes:

$$u^k = u^{k-1} + M_{RAS,\delta}^{-1}\left(f - Au^{k-1}\right), \text{ with } M_{RAS,\delta}^{-1} = \sum_{i=1}^{p} \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta} \tag{4}$$

When the number of subdomains increases the convergence rate of RAS decreases. When it is applied to linear problems, the RAS has a pure linear rate of convergence.

First we study the robustness and scalability of the preconditioner on a $2D$ driven cavity problem. Second we propose a test of the quality of our coarse space on an $2D$ industrial problem.

3.1 Robustness

Here, we want to study the numerical scalability of the method for the domain decomposition preconditioner chosen. We fix the number of Richardson iteration to perform while decreasing the convergence rate of the preconditioner, *i.e.* we set a coarse space size l and increases the number of partitions p.

We consider a test case called $e30r2000$ coming from modeling 2D fluid flow in a driven cavity proposed in the Matrix Market data collection [3] referenced under the name DRIVCAV. The flow is modeled using the incompressible Navier Stokes equations discretized using Finite Element Method and linearized using Newton's method. The unit square on what the problem is solved is dicretized by 30 elements on the edges. The Reynolds number is set to 2000.

The matrix A is real, non-symmetric and indefinite of size $m = 9661$ and has 306356 entries. The estimated condition number given by the condest function of MATLAB is $\kappa_\infty(A) = 6.77e+11$.

We partition the operator with the METIS software for partitioning graphs with a multilevel recursive-bisection algorithm, in $p = \{4, 8, 12\}$ partitions. We compute $l = 60$ iterations of a RAS iterative process starting from an initial guess $u^0 = 0$, and perform the SVD of the corresponding sequence of vectors.

Figure 1 (top) shows the singular values profile. When p increases the spectrum coverage decreases which implies a decreasing of the quality of the solution on the coarse space.

Figure 1 (bottom) shows the convergence to the solution of a GCR method preconditioned by a RAS preconditioning technique on the left and enhanced by the given algebraic two-level approach with initialisation of the Krylov method by the solution of the coarse system written on $\mathbb{R}^m$. The convergence rate of the RAS method is reduced for each choice of partitioning. For $p = 4$ and $p = 8$ the initialization by the coarse solution is efficient and we observe an enhancement about 8 and 2 orders of convergence at the initialization respectively. For all partitioning the accuracy is better than for the RAS, *i.e.* the GCR reaches greater convergences and, although there is still a plateau due to the bad conditioning of the system, the convergence to the solution for $p = 12$ can reach 10^{-7} instead of 10^{-5}.

3.2 Quality

Here, we want to observe the influence of the quality of the coarse space on the convergence rate of the preconditioned solution method.

We apply our technique on the case GT01R proposed by a CFD company called FLUOREM, on [1], which deals with steady flow parametrization. From a steady RANS simulation (compressible Navier-Stokes equations) on a reference configuration they obtain linear systems with real, square and indefinite matrices. Those matrices, generated through automatic differentiation of the flow solver around a steady state, correspond to the Jacobian with respect to the conservative fluid variables of the discretized governing equations (finite-volume discretization). The right hand side represents the derivative of the equations with respect to a parameter (of operation or shape).

The CASE_004 GT01 operator comes from a $2D$ inviscid case in the context of a linear cascade turbine. The solution of the discrete system is defined over five variables per node. The discretisation is done among 1596 nodes, describing one inter-blade channel. The stencil involved by the convective scheme uses nine nodes. Thus, there are nine non-zero blocks for each node in the matrix. The peculiarity is that the computational domain is periodic, which introduces some non-zero elements far away from the diagonal. The resulting matrix is real, non-symmetric and not positive definite, of size $m = 7980$.

Figure 2 shows the singular values (left) obtained after 20, 40 or 60 iterations of a RAS iterative process with $p = 8$. For $l = 60$, σ covers 15 orders of magnitude, while it covers 10 orders of magnitude for $l = 40$ and 5 for $l = 20$. For each we

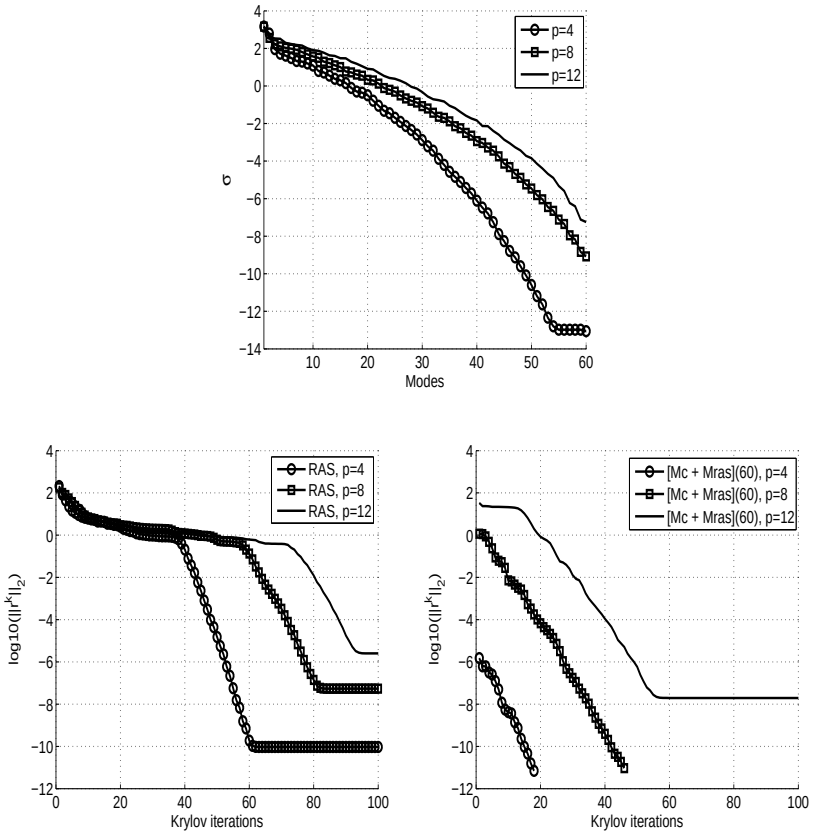


Fig. 1 Solving 2D driven cavity, Re=2000, n=9661, with GCR preconditioned by RAS (left) and ML_RAS_svd(60) (right). Singular values of RAS solutions to compute M_c^{-1} for p = 4, 8, 12 (top).

choose $l = q$. As expected, the convergence of the GMRES (right) is better when q increases. Nevertheless, the convergence plots for 20 and 40 singular values kept are similar.

Table 1 shows the coarse solution accuracy compared to a solution given using LU factorization. The greater the number of iterations of a Richardson process is, the better the coarse solution accuracy is.

Those results shows that, although the quality of the coarse space is increasing with the number of Richardson iterations, it is not necessary to compute a lot of singular values to enhance the convergence with this technique.

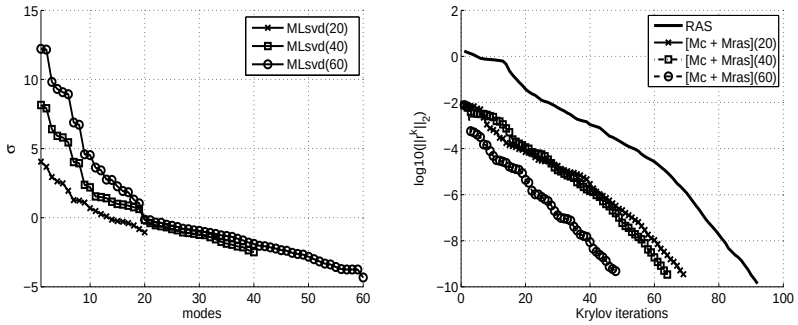


Fig. 2 Solving 2D case, GT01, n=7814, with GMRES preconditioned by RAS and ML_RAS_svd(q), p = 8 (right), Singular values of RAS solutions to compute M_c^{-1} (left)

modes	20	40	60
$\|u_{ex} - \mathbb{U}_q^T u_c\|$	7.23 e-01	5.99 e-02	8.39 e-03

Table 1 Coarse solution accuracy for the GT01 case, compared to a solution given using *LU* factorization.

4 Conclusion

As in [9] and [6] the principle of using the SVD of successive solutions of an iterative process enables to compute a coarse solution without the knowledge of the underlying equations but it not used to accelerate a sequence of vectors but to construct a Krylov subspace. Then it can also be used to construct algebraic coarse levels for a two-level preconditioning technique based on any preconditioner which can be used in an iterative Richardson process.

Numerical results have been shown for the RAS preconditioning technique on two fluid flow problems. The algebraic framewok enables to deal with real, non-symmetric and not positive definite operators. The two-level preconditioners produced are numerically scalable for domain decomposition technique such as RAS and the coarse space enables to compute an approximation of the solution which is used to initialize the chosen Krylov method.

Further work concerns the study of the non-singularity of the coarse operators built with this approach [4]. Moreover, a discussion about the choice of the SVD algorithms and the quality of the coarse space produced should be studied.

References

1. Boisvert, R.F., Pozo, R., Remington, K., Barrett, R.F., Dongarra, J.J.: Matrix market: A web resource for test matrix collections. In: The Quality of Numerical Software: Assessment and Enhancement, pp. 125–137. Chapman & Hall (1997). URL http://math.nist.gov/MatrixMarket
2. Cai, X.C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J. Sci. Comput. **21**(2), 792–797 (electronic) (1999). DOI 10.1137/S106482759732678X. URL http://dx.doi.org/10.1137/S106482759732678X
3. Davis, T.A., Hu, Y.: The university of florida sparse matrix collection. ACM Transactions on Mathematical Software **38**(1) (2011) URL http://www.cise.ufl.edu/research/sparse/matrices
4. Dufaud, T.: An algebraic two-level methodology based on information of a richardson iterative process for constructing preconditioners (20YY). In preparation
5. Dufaud, T., Tromeur-Dervout, D.: Aitken's acceleration of the restricted additive Schwarz preconditioning using coarse approximations on the interface. C. R. Math. Acad. Sci. Paris **348**(13-14), 821–824 (2010). DOI 10.1016/j.crma.2010.06.021. URL http://dx.doi.org/10.1016/j.crma.2010.06.021
6. Dufaud, T., Tromeur-Dervout, D.: Efficient parallel implementation of the fully algebraic preconditioning technique aras2. Advances in Engineering Software (20YY). Submitted
7. Eisenstat, S.C., Elman, H.C., Schultz, M.H.: Variational iterative methods for nonsymmetric systems of linear equations. SIAM J. Numer. Anal. **20**(2), 345–357 (1983). DOI 10.1137/0720023. URL http://dx.doi.org/10.1137/0720023
8. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **7**(3), 856–869 (1986). DOI 10.1137/0907058. URL http://dx.doi.org/10.1137/0907058
9. Tromeur-Dervout, D.: Meshfree adaptative aitken-schwarz domain decomposition with application to darcy flow. In: B. Topping, P. Ivanyi (eds.) PARALLEL, DISTRIBUTED AND GRID COMPUTING FOR ENGINEERING, *Computational Science Engineering and Technology Series*, vol. 21, pp. 217–250. SAXE-COBURG PUBLICATIONS (2009)

Part III
Contributed Presentations

Distributed Nonsmooth Contact Domain Decomposition (NSCDD): algorithmic structure and scalability

V. Visseq[1], A. Martin[2], D. Dureisseix[3], F. Dubois[1], and P. Alart[1]

1 Introduction

Numerical simulations of the dynamics of discrete structures in presence of numerous impacts with frictional contacts leads to CPU-intensive large time computations. To deal with these problems (e.g. granular materials, masonry structures), numerical tools have been developed, such as the nonsmooth contact domain decomposition (NSCDD), presented Sec 2. We focus herein on a distributed version with parallel detection of fine contacts (Sec. 3) and on two possible communication schemes to solve the interface problem (Sec. 4). Those improvements allow to study scalability and numerical performances of the method for 2D and 3D granular media (Sec. 5).

2 The nonsmooth contact domain decomposition

2.1 Nonsmooth contact dynamics reference problem

In this section we recall briefly the background theory of nonsmooth contact dynamics in the context of time-stepping schemes before an analysis of the main steps of the NSCDD method.

With a time-stepping scheme, no event detection is performed. Once the solution is known at the beginning of a time slab $[t_i, t_{i+1}]$, whose known quantities are denoted with a superscript $(-)$, the quantities at the end of the time slab (without a superscript) have to be determined.

Grain nonsmooth dynamics. Considering a rigid model for the grains, the dynamics of the granular medium is written as the vector equation [4]:

[1] LMGC – UMR 5508, Université Montpellier II / CNRS, CC 048 Place Eugène Bataillon, F-34095 Montpellier Cedex 5, France e-mail: {FirstName}.{Name}@univ-montp2.fr .[2] LaMSID – UMR 8193 EDF / CNRS / CEA , EDF R&D, 1 avenue du Général de Gaulle, F-92141 Clamart Cedex, France e-mail: alexandre-externe.martin@edf.fr .[3] David Dureisseix LaMCoS – Université de Lyon, UMR 5259 INSA Lyon / CNRS, Bâtiment Jean d'Alembert, 18-20 rue des Sciences, F-69621 Villeurbanne Cedex, France e-mail: David.Dureisseix@insa-lyon.fr

$$MV - R = R^d, \tag{1}$$

where the prescribed right-hand side is $R^d = R^D + MV^-$. V is the velocity of the grain (it contains the translational degrees of freedom – dof, and the rotational ones); R is the resultant impulse on the grain due to interactions with other grains and R^D are the external prescribed impulses. The matrix M contains both the mass (for the translational dof) and the inertia (for the rotational dof). The assembly of these equations (independent for each grain) is formally written in the same way (1).

Contact interaction. For a unilateral contact Moreau proved via a viability lemma [4], that we can use a velocity-impulse complementary law:

$$\mathscr{R}(v,r) = 0, \tag{2}$$

v is the velocity jump at the contact point between the two interacting grains, r is the impulse at the same contact point. $\mathscr{R}$ is usually a non linear and multivalued relationship between the previous two dual quantities. Both v and r are expressed in the local coordinate basis to the contacts between the interacting grains. Therefore, they are linked to the global kinematic and static quantities with compatibility conditions $v = H^T V$ and $R = Hr$.

Reduced dynamics. Taking the dynamics (1) and the compatibility conditions into account, the reduced dynamics involving material variables can be obtained:

$$Wr - v = -v^d, \tag{3}$$

where W is the Delassus operator: $W = H^T M^{-1} H$, and $v^d = H^T M^{-1} R^d$. To close the problem, one adds the constitutive relation (2), and the reference problem reads:

$$\begin{cases} Wr - v = -v^d \\ \mathscr{R}(v,r) = 0 \end{cases}. \tag{4}$$

The difficulty to solve this problem is at least two-folds: on one hand, the number of unknowns (number of interaction quantities r and v) may be large (for instance, an average of $6.5\ 10^5$ unknowns for the 3D problem illustrating this paper), and the Delassus operator W is not well conditioned. On the other hand, the constitutive relation is nonsmooth (e.g. it is non linear, and not differentiable). To address the nonsmoothness issue, the NSCD (nonsmooth contact dynamics) method with a non-linear Gauss-Seidel (NLGS) solver [4, 2] is used. To address the large size of the problem, a substructuring approach is proposed.

2.2 Sub-structuring

The proposed sub-structuring may be seen as a FETI-like domain decomposition. Indeed, after the partition of the sample (step detailed in section 3) constraints are

added on the interface grain velocities, with E the index of a subdomain:

$$\sum_{\mathrm{E}=1}^{n_\mathrm{s}} A_{\Gamma\mathrm{E}} V_\mathrm{E} = 0, \tag{5}$$

n_s is the number of subdomains, $A_{\Gamma\mathrm{E}}$ is a signed boolean matrix selecting interface grains among subdomains to construct their velocity jumps. This step consists of a perfect gluing procedure, which is quite different from the approach proposed in [3]. The dynamics per subdomain reads:

$$M_\mathrm{E} V_\mathrm{E} - R_\mathrm{E} = R_\mathrm{E}^d - A_{\Gamma\mathrm{E}}^T F_\Gamma, \tag{6}$$

where F_Γ are the Lagrange multipliers associated to the previous constraints. One shows that combining equation (5) and (6) the interface problem reads:

$$X F_\Gamma = \sum_{\mathrm{E}=1}^{n_\mathrm{s}} A_{\Gamma\mathrm{E}} M_\mathrm{E}^{-1} \left(R_\mathrm{E} + R_\mathrm{E}^d \right), \tag{7}$$

with $X = \sum_{\mathrm{E}=1}^{n_\mathrm{s}} A_{\Gamma\mathrm{E}} M_\mathrm{E}^{-1} A_{\Gamma\mathrm{E}}^T$ the interface operator [5]. The reduced dynamics problem per subdomain has the same structure that the global one provided the addition of Lagrange multipliers as additional external impulses on the given right hand side:

$$\begin{cases} W_\mathrm{E} r_\mathrm{E} - v_\mathrm{E} = -v_\mathrm{E}^d + v_\mathrm{E}^\Gamma \\ \mathscr{R}(v_\mathrm{E}, r_\mathrm{E}) = 0, \end{cases}, \tag{8}$$

where $v_\mathrm{E}^\Gamma = H_\mathrm{E}^T M_\mathrm{E}^{-1} A_{\Gamma\mathrm{E}}^T F_\Gamma$. To close the problem, the interface behavior (5) or (7) should be added.

2.3 NSCDD algorithmic structure in the LMGC90 platform

The NSCDD method has been implemented into the LMGC90 platform[1] [1] for time-evolution problems (N is the number of time steps). Algorithm 1 describes its structure. A NSCDD iteration is then composed of n_GS Gauss Seidel iterations on the reduced dynamics and an update of interface quantities. In practice n_GS is chosen equals to 1. In the next two sections we will focus on the underlined stages (with boldface) in the following algorithm 1.

[1] www.lmgc.univ-montp2.fr/LMGC90

Algorithm 1 NonSmooth Contact Domain Decomposition (NSCDD)

for $i = 1,\dots,N$ **do**
  Contact detection (eventually parallelized) and possible new decomposition of the domain
  Initialize unknowns at time t_i: (r_E, v_E, F_Γ)
  while (convergence criterion not satisfied) **do**
    In parallel for $E = 1,\dots,n_s$:
      Compute the velocity $\bar{v}_E^\Gamma$
      Compute $(\bar{r}_E,\bar{v}_E)$ with n_{GS} non-linear Gauss-Seidel iterations on:

$$\begin{cases} W_E \bar{r}_E - \bar{v}_E = -\bar{v}_E^{fl} + \bar{v}_E^\Gamma \\ \mathscr{R}(\bar{v}_E, \bar{r}_E) = 0 \end{cases} \tag{9}$$

      Update $(r_E, v_E) \leftarrow (\bar{r}_E, \bar{v}_E)$
      Compute R_E and correct the velocity on interface grains to get $A_{\Gamma E}\bar{V}_E$
    In sequential, but may be possibly parallelized (DCS version):
      Compute ΔF_Γ **as:** $X\Delta F_\Gamma = \sum_{E=1}^{n_s} A_{\Gamma E}\bar{V}_E$ **and update interface impulses** F_Γ
  end while
  Update grain positions in parallel
end for

3 Contact detection

At the beginning of a time step, positions and velocities of grains are known and the contact network between bodies has to be computed. Contact detection is a CPU time consuming task, especially for a large number of bodies —this is directly related to the number and the shape of the elements considered. Usually, an efficient solution is to proceed to a two-level detection, i.e. a rough (and cheap) detection followed by an elimination of loose contact predictions and the computation of contact frame (the fine detection).

3.1 Partitioning based on "rough" contact network

Once a rough detection has been performed, the interaction graph consists in nodes associated to grains and edges associated to interactions. We choose to distribute interactions among subdomains as in [5] (we proceed by distributing the middle points between the centers of mass of interacting grains, according to their coordinates, using an arbitrary regular underlying grid, Figure 1(a)). Indeed, with such a choice, the "boundary" grains are duplicated in the subdomains. If a grain indexed with i is connected with m^i subdomains, m^i is called its multiplicity number. For consistency for the rigid model of the grains, the masses and moments of inertia are distributed among the neighboring subdomains according to their multiplicity number, in a partition of unity manner. We remark that rough detection, and so the domain partitioning, does not have to be done at each time step, but at a user-defined frequency (fixed at 10 time steps for numerical tests of section 5).

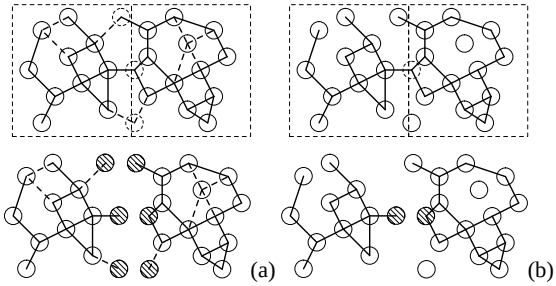


Fig. 1 Rough (a) and fine (b) interaction network and their associated domain partitioning. Striped grains represent grains of multiplicity $m^i > 1$; dashed lines represent interactions roughly detected which vanishes in effective contact network.

3.2 Parallelized fine detection

Once the domain decomposition has been performed, data can be distributed among the processors and a fine contact detection can be performed in parallel on each substructure local data. Nevertheless contacts roughly detected may disappear at the end and the multiplicity number of the grains may have been incorrectly predicted (Figures 1(a) and 1(b) show cases we may encountered). In particular, predicted boundary grains could turn out not to belong to the minimal interface (computed thanks to the fine contact graph). Their adding to the interface gluing step does not change the problem to solve but increases the size of data to transfer between processors. A future optimization should be to correct interface structures and material parameters to take this phenomenon into account.

4 Communication schemes for solving interface problem

In this section we present two communication schemes associated to centralized and distributed interface problem solving procedure. As one has to solve the interface problem for each NSCDD iterations, to define an appropriate algorithmic formulation, minimizing the data exchanges between processes, is a key issue for the performances of the proposed method.

4.1 Centralized communication scheme (CCS)

At a first glance, the interface gluing step (7) is defined as a global linear equation linking all the subdomains. This is replaced in the iterative algorithm by requiring communications between the subdomains such that one process gathers all the velocity contributions to the vector of velocity jumps. The value of the Lagrange multipliers F_Γ computed sequentially is then distributed such that subdomain E receives its minimal data amount $A_{\Gamma E}^T F_\Gamma$.

4.2 Decentralized communication scheme (DCS)

Due to the structure of the interface operator X, extensively studied in [5], each distributed database (per process related to subdomain E^*) is sufficient to construct the elementary contribution to the interface operator:

$$X_{\Gamma_{\mathrm{E}^*}} = \sum_{\mathrm{E}=1}^{n_\mathrm{s}} A_{\Gamma_{\mathrm{E}^*}\mathrm{E}} M_\mathrm{E}^{-1} A_{\Gamma_{\mathrm{E}^*}\mathrm{E}}^T, \tag{10}$$

$A_{\Gamma_{\mathrm{E}^*}\mathrm{E}}$ is a signed boolean matrix, mapping grains of subdomain E to velocity jumps of the elementary interface Γ_{E^*} (restriction of the global interface to the boundary of subdomain E^*). Then, an elementary interface problem can be defined as:

$$X_{\Gamma_{\mathrm{E}^*}} \Delta F_{\Gamma_{\mathrm{E}^*}} = \sum_{\mathrm{E}=1}^{n_\mathrm{s}} A_{\Gamma_{\mathrm{E}^*}\mathrm{E}} V_\mathrm{E}. \tag{11}$$

Finally, the data gathering of $\sum_{\mathrm{E}=1}^{n_\mathrm{s}} A_{\Gamma_{\mathrm{E}^*}\mathrm{E}} V_\mathrm{E}$ on each process corresponds to data exchanges over an unstructured topology. Indeed discrete element methods, such contact dynamics, may deal with large/elongated bodies, possibly related to all subdomains. A common example of such bodies is a wall which support contacts on a large range. With the computation of the assembling of local contributions, it is easy to show that this is the expected iterated vector:

$$\Delta F_\Gamma = \sum_{\mathrm{E}=1}^{n_\mathrm{s}} B_{\Gamma\mathrm{E}} D_\mathrm{E} B_{\Gamma_\mathrm{E}\mathrm{E}}^T \Delta F_{\Gamma_\mathrm{E}}, \tag{12}$$

$B_{\Gamma\mathrm{E}}$ is a boolean matrix selecting interface grains among subdomains, $B_{\Gamma_\mathrm{E}\mathrm{E}}$ is a boolean matrix selecting elementary interface grains among subdomains and D_E is a diagonal matrix with value $1/m^i$ for entries corresponding to grain i.

4.3 Performance comparison of the two communication schemes

The influence of the proposed communication schemes is studied regarding the CPU time percentage consumed during MPI exchanges (Table 1) with respect to the whole CPU time of a simulation. The test consists of a sample with 55000 spheres submitted to an isotropic compaction, over 500 time steps (Figure 2).

Results presented in Table 1 show clearly the gain we may obtain considering DCS compared to CCS. Decentralized communication scheme indeed allows to avoid MPI collective communications (especially expensive, in our case, to scatter updating of Lagrange multipliers) and to partially parallelize interface treatment.

Table 1 Comparison of elapsed CPU time percentage consumed during MPI exchanges for centralized (CCS) and decentralized (DCS) communication schemes; isotropic compaction of a 55000 spheres sample.

n_s	Partitioning parameters (x,y,z)	CPU percentage (CCS)	CPU percentage (DCS)
1	1 1 1	0 %	0 %
3	3 1 1	31.3 %	14.0 %
4	2 2 1	35.6 %	9.1 %
8	2 2 2	58.3 %	18.4 %

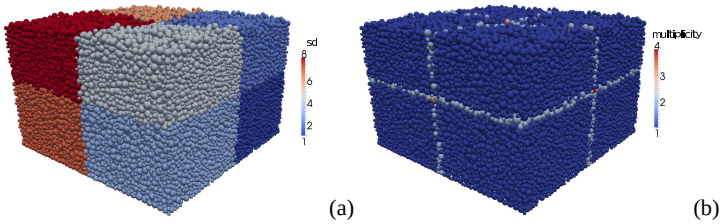


Fig. 2 Sample of 55000 spheres submitted to isotropic compaction. Subdomains indexes (a) and multiplicity number of grains (b).

5 Scalability preliminary results

We propose to study scalability of the NSCDD method on tests consisting in samples of (2D) disks and (3D) spheres submitted to basic loadings. The speedup S_p, function of the number of processes N_p (supposed equals to the number of subdomains), and the number of total iterations, over 100 time steps, are then highlighted. On both tests, friction is considered at contact between particles. Simulations are performed on a 48 cores AMD processor.

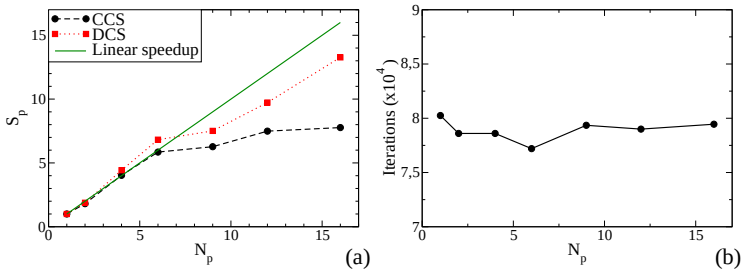


Fig. 3 Speedup (a) and total number of iterations (b); biaxial loading of a 13000 disks sample.

2D – biaxial test. As shown in Figure 3, the speedup does not change drastically depending on the communication scheme for a quite small 2D sample, at least

for a small number of processes. The number of iterations (independent from the communication scheme) is nearly constant for all the tested domain splittings.

3D – triaxial test. For 3D granular samples (Figure 4) the centralized communication scheme has very poor efficiency so it is not reported here. We consider a random closed packing of 64000 spheres subjected to triaxial compaction (downward displacement of the top wall with a constant velocity and confining stress acting on the lateral walls). That is the hardest mechanical configuration one may encountered because of the strong indeterminacy of the problem cumulated to the high number of contacts unknowns (6.5×10^5 in average in our case), but also the most interesting numerical case for the domain decomposition method proposed. We see that the speedup has good quantitative behavior, even if the hardware and MPI library optimization may be improved. Indeed, the use of about a hundred processors (for larger problems than those studied here) implies to mobilize a supercomputing platform to obtain reasonable speedup.

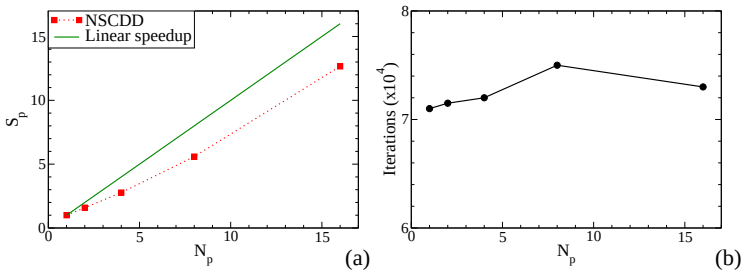


Fig. 4 Speedup (a) and mean number of iterations (b); 64000 spheres sample.

Acknowledgements This work was partly supported by OSEO, FEDER and the region of Languedoc-Roussillon (Degrip project).

References

1. Dubois, F., Jean, M., Renouf, M., Mozul, R., Martin, A., Bagneris, M.: LMGC90. In: 10e colloque national en calcul des structures. Giens, France (2011)
2. Jean, M.: The non-smooth contact dynamics method. Comput Method Appl M **177**, 235–257 (1999)
3. Koziara, T., Bićanić, N.: A distributed memory parallel multibody contact dynamics code. Int J Numer Meth Eng **87**(1-5), 437–456 (2011)
4. Moreau, J.J.: Numerical aspects of sweeping process. Comput Method Appl M **177**, 329–349 (1999)
5. Visseq, V., Martin, A., Iceta, D., Azema, E., Dureisseix, D., Alart, P.: Dense granular dynamics analysis by a domain decomposition approach. Comput Mech **49**, 709–723 (2012)

Integrating an N-body problem with SDC and PFASST

Robert Speck[1], Daniel Ruprecht[1,5], Rolf Krause[1], Matthew Emmett[2], Michael Minion[3], Mathias Winkel[4], and Paul Gibbon[4]

1 Introduction

Particle methods are an attractive approach for solving complex three-dimensional flow problems since they are naturally adaptive [4]. In this work, we utilize a particle description based on vorticity to discretize the Navier-Stokes equations in space, which results in a first-order initial value ODE for the particles' positions and vorticities. When highly accurate solutions to the initial value problem are required, it is usually more efficient to use higher-order temporal integration schemes. Spectral Deferred Correction (SDC) methods [6] are an elegant way to achieve high-order time integration by using simple low-order schemes in an iterative fashion. While a single time-step of an SDC method is usually more expensive in terms of computation time than a step of a classical Runge-Kutta scheme, SDC can be competitive in terms of time-to-solution required for a fixed accuracy [3].

The temporal integration in vortex methods requires the evaluation of an N-body problem for each function evaluation. This evaluation can be parallelized efficiently by a spatial distribution of the particles over multiple cores. However, the strong scalability of this approach is limited when the number of degrees-of-freedom per core becomes too small (similar to domain decomposition techniques for mesh-based methods). Time-parallel methods are one possible approach to speed up simulations beyond this saturation point, with early approaches dating back to [18]. A very general scheme is Parareal [15], which allows arbitrary integration schemes to be used in a black-box fashion. A detailed mathematical analysis of Parareal is conducted in [8] and comprehensive lists of references can be found e. g. in [17, 20]. The drawback of Parareal is that the parallel efficiency is formally bounded by $1/K$ where K is the number of iterations required for convergence. The *Parallel Full Approximation Scheme in Space and Time* (PFASST) method for parallelizing SDC methods in time is introduced in [7, 17]. By combining the iterations of SDC with the iterations of Parareal, it significantly relaxes the efficiency bound of Parareal and further enhances the competitiveness of integration schemes based on SDC.

[1]Institute of Computational Science, Università della Svizzera italiana, Lugano, Switzerland, e-mail: {robert.speck,daniel.ruprecht,rolf.krause}@usi.ch · [2]Lawrence Berkeley National Laboratory, Berkeley (USA), e-mail: mwemmett@lbl.gov · [3]Institute for Computational and Mathematical Engineering, Stanford University, Stanford (USA), e-mail: mlminion@gmail.com · [4]Jülich Supercomputing Centre, Jülich (Germany), e-mail: {m.winkel,p.gibbon}@fz-juelich.de · [5]Mathematisches Institut, Heinrich-Heine-Universität Düsseldorf, Düsseldorf (Germany).

The present paper investigates the accuracy of integrating a particle-based discretization of the 3D Navier-Stokes equations in time using SDC and PFASST. We are not aware of any other studies that investigate SDC integration methods in conjunction with particle-based spatial solvers aside from a small, one-dimensional N-body example in [2] for *Revisionist Integral Deferred Corrections* (RIDC).

Since this work focuses on the accuracy of the temporal discretization, the N-body problem is solved directly with $\mathcal{O}(N^2)$-complexity, limiting the presented studies to rather small numbers of particles. The unfavorable quadratic complexity can be overcome by computing approximate interactions using e. g. Barnes-Hut tree codes [1] or the Fast Multipole Method [11]. Results on the strong scaling of PFASST on extreme scales, simulating merely 4 million particles on up to 262,144 cores, are reported in [26], where the massively parallel Barnes-Hut tree code PEPC [9, 10, 23, 24, 27] is applied. There, however, only a very brief discussion of accuracy is given, aiming solely at identifying parameters that generate time-parallel and time-serial solutions of comparable quality that allow for a meaningful comparison in terms of runtimes. Here, accuracy of the method is addressed in more detail, including a comparison with a standard Runge-Kutta scheme.

We briefly describe SDC and PFASST in Section 2 and present accuracy studies in Section 3. In Section 4 we summarize our results and comment on how further efficiency can be achieved for particle-based methods as a prelude to the large-scale simulations performed in [26].

2 Parallel in Time Integration using Spectral Deferred Corrections

Discretizing the vorticity-velocity formulation of the Navier-Stokes equations with N particles results in an initial value problem of the form (see e. g. [4])

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) = f(\mathbf{y}(t)), \quad \mathbf{y}(0) = \mathbf{y}_0 \in \mathbb{R}^{6N}, \quad t \in [0,T] \tag{1}$$

where the right-hand side f denotes the sum of all mutual particle interactions (commonly via regularized smoothing kernels) and $\mathbf{y} \in \mathbb{R}^{6N}$ is a vector containing $3D$ positions and vorticities of N particles. Without further approximation, directly evaluating f is of $\mathcal{O}(N^2)$-complexity. This can be significantly reduced using multipole approximations with either Barnes-Hut tree-codes [22] or the Fast Multipole Method [5] at the cost of increased spatial approximation errors. In this work, however, we focus on errors from temporal discretization, and hence f is evaluated exactly with full accuracy and quadratic complexity. To solve the initial value problem (1), classical explicit time-stepping algorithms such as fourth-order Runge-Kutta schemes are commonly used, see e. g. [19]. Here, we use SDC methods [6], which can easily produce high-order time integration schemes from simple low-order methods and which can be parallelized in time using PFASST [7, 17].

Let $0 = t_0 < t_1 < t_2 < \ldots < t_M = T$ denote a discretization of the time-interval $[0,T]$, and let $t_m \leq \tau_1^m < \tau_2^m < \ldots \tau_J^m \leq t_{m+1}$ denote a set of quadrature points in the interval $[t_m, t_{m+1}]$, see [14] for details on the choice of these nodes. For brevity, we fix m and write τ_j instead of τ_j^m for all $j = 0, \ldots, J$. Moreover, let $\mathbf{y}_j$ denote an approximation to $\mathbf{y}(t_j)$, $j = 0, \ldots, J$. Starting from the equivalent Picard formulation of Eq. (1), the key ingredient of SDC is the spectral approximation

$$\mathbf{S}_j^{j+1} f = \sum_{l=0}^{J} \alpha_{j,l} f(\mathbf{y}_l) \approx \int_{\tau_j}^{\tau_{j+1}} f(\mathbf{y}(\tau)) \, \mathrm{d}\tau \tag{2}$$

with quadrature weights $\alpha_{j,l} \in \mathbb{R}$. Then, the $k+1$ explicit update for $\mathbf{y}_{j+1}$ at node $j+1$ using low-order explicit Euler is evaluated as

$$\mathbf{y}_{j+1}^{k+1} = \mathbf{y}_j^{k+1} + \Delta\tau_j \left[f(\mathbf{y}_j^{k+1}) - f(\mathbf{y}_j^k) \right] + \mathbf{S}_j^{j+1} f^k, \tag{3}$$

where $\Delta\tau_j = \tau_{j+1} - \tau_j$, k is the iteration index and $\mathbf{y}_j^0$ is some provisional solution computed at the nodes τ_j. For K iterations with a first-order propagator, SDC formally results in a Kth-order time integrator, provided the quadrature approximation is accurate enough. On the other hand, using M Gauss-Lobatto quadrature nodes yields a method of order $2M-2$, provided the number of iterations K is large enough. We refer to [6, 12] for more details and properties of this approach.

To introduce parallel time-stepping we briefly review the Parareal approach [15]. Here, temporal parallelization of (1) is imposed by iterating over two integration schemes, a fast and inaccurate one (the "coarse" propagator) denoted typically as $\mathscr{G}$ and a slow but accurate one (the "fine" propagator) labeled $\mathscr{F}$. While a classical time-marching scheme computes a sequence of solutions

$$\mathbf{y}_{m+1} = \mathscr{F}(\mathbf{y}_m), \; t_m \in [0,T], \tag{4}$$

Parareal replaces (4) by the iteration

$$\mathbf{y}_{m+1}^{k+1} = \mathscr{G}(\mathbf{y}_m^{k+1}) + \mathscr{F}(\mathbf{y}_m^k) - \mathscr{G}(\mathbf{y}_m^k), \; m = 0, \ldots, M-1 \tag{5}$$

where $k \geq 0$ is again the iteration index. The key here is that if the solution from iteration k is known, the expensive evaluation of the terms $\mathscr{F}(\mathbf{y}_m^k)$ can be done in parallel for multiple m. Then, a correction is propagated from t_0 to t_M through the cheap yet serial computation of the terms $\mathscr{G}(\mathbf{y}_m^{k+1})$. The Parareal iteration (5) converges to a solution of the same accuracy as obtained by running $\mathscr{F}$ in serial (i. e. by computing (4)). For N_{it} iterations of Parareal and N_{p} processors, the speedup achievable is bound by $N_{\mathrm{p}}/N_{\mathrm{it}}$, see [17].

To improve parallel efficiency, the PFASST algorithm intertwines SDC integrators of different accuracy for $\mathscr{G}$ and $\mathscr{F}$ with the iterations of Parareal. In addition to multiple levels in time, PFASST can benefit from spatial coarsening, as used in e. g. multi-grid techniques. Furthermore, PFASST employs Full Approximation Scheme (FAS) corrections to increase the accuracy of SDC iterations

on coarse levels. Many details of SDC, Parareal and PFASST have been omitted here for brevity, and the reader is referred to the more detailed discussions in e. g. [6, 7, 15, 17, 20, 26].

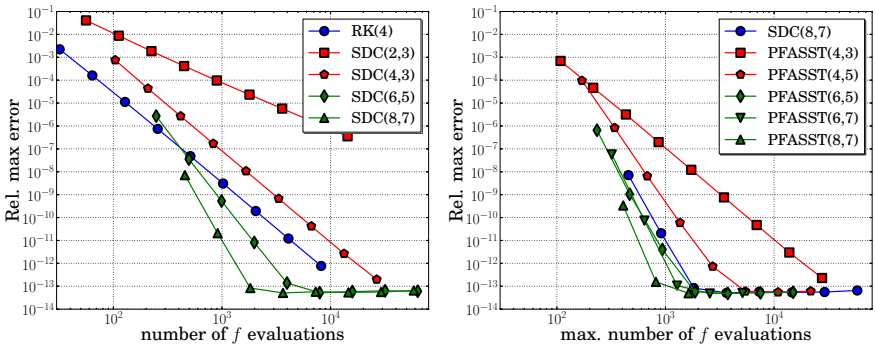
3 Numerical Results

To test SDC and PFASST in the framework of particle simulations we use a standard spherical vortex sheet setup as discussed e. g. in [21, 25] with $N = 10{,}000$ particles (i. e. 60,000 degrees-of-freedom), a sixth-order algebraic kernel for the regularization, and final time $T = 32$. For convenience, the direct evaluation of f has been parallelized in space using 64 processors of the Intel Cluster JUROPA at Jülich Supercomputing Centre [13]. Reported errors are computed using a reference solution generated by an 8th-ordered SDC method with 2,048 time-steps.

Figure 1(a) shows the relative maximum error against the number of evaluations of f for the standard RK4 method (denoted RK(4)) and SDC with different numbers of iterations and Gauss-Lobatto quadrature nodes (denoted SDC(X,Y) for X iterations and Y Gauss-Lobatto nodes). The left-most markers correspond to 8 time-steps, the rightmost to 2,048. Here, the order of convergence of SDC equals the number of performed iterations, since the number of quadrature nodes is high enough, see [6]. Increasing the order of SDC (i. e. increasing the number of iterations) reduces the error substantially for a given number of function evaluations. Hence, if solutions of moderate or high accuracy are sought, e. g. below 10^{-7}, higher-order SDC methods are more efficient in terms of f evaluations than RK4 or lower-order SDC methods.

Although fourth-order SDC (realized here by SDC$(4,3)$ in Figure 1(a)) is more expensive than the classical RK4, one advantage of SDC methods is that the order of convergence can be easily controlled by the interplay of quadrature nodes and iterations. Implementing Runge-Kutta schemes of higher-order, on the other hand, involves tedious and error-prone code re-implementations. Moreover, SDC can easily treat stiff and non-stiff parts of the right-hand side separately and/or with differing time-steps accuracy [16]. More importantly for the present study is that SDC methods can be parallelized in time using PFASST.

In Figure 1(b), we show the relative maximum error of two-level PFASST runs on four time-processes, i. e. with four times more processors than the space-parallel/time-serial SDC runs. Here, the x-axis depicts the maximum number of f evaluations performed by one process (maximum is typically attained on the last time-rank), counting evaluations of f in $\mathscr{F}$ as well as in $\mathscr{G}$, see also the discussion on Figure 2 below. We do not employ spatial coarsening, i. e. the propagators $\mathscr{F}$ and $\mathscr{G}$ differ only in the number of temporal quadrature nodes. We also compare the PFASST runs to a serial SDC run with 8 iterations on 7 Gauss-Lobatto nodes. PFASST, while being more expensive in terms of f evaluations, also yields a higher accuracy for a given number of iterations and quadrature nodes in our case. This is due to the fact that each iteration contains sweeps at both fine and coarse levels, so



(a) SDC vs. RK(4) (b) PFASST vs. SDC$(8,7)$

Fig. 1 Error versus number of evaluations of the right-hand side f for SDC(X,Y) and the maximum number of f evaluations performed by one process (typically on the last time-rank) for PFASST(X,Y) on 4 time-processes. Direct particle simulation of a spherical vortex sheet with 10,000 particles, sixth-order algebraic kernel, and up to 2,048 time-steps.

that effective number of SDC sweeps is higher than the number of iterations. Note again that higher-order schemes show much better efficiency in terms of accuracy versus f evaluations.

Besides providing higher accuracy, PFASST also introduces an additional layer of parallelism. Provided that the spatial parallelization is already saturated, the application of PFASST can push the strong-scaling limit further by distributing the temporal integration across multiple time-processes, as shown in [26]. To shed more light on this concept, Figure 2 shows the number of f evaluations required by PFASST$(8,7)$ on one to eight time-processors with 16 time-steps. As a reference, we choose SDC with 12 iterations on 7 quadrature nodes to obtain a comparable accuracy to PFASST$(8,7)$ when tested against a very fine resolved reference solution: Both schemes provide an accuracy of approx. 10^{-13} with 16 time-steps. The number of f evaluations in $\mathscr{G}$ are highlighted in blue and hatched, the ones in $\mathscr{F}$ in red. Note that there is no coarse propagator in the single-level SDC scheme. The second bar, denoted $P(8,7,1)$, corresponds to PFASST with 8 iterations and 7 quadrature nodes on one time-processor. When run on one time-processors, PFASST reduces to a multi-level SDC scheme. Comparing SDC$(12,7)$ and PFASST$(8,7,1)$ we note that by switching from a single-level SDC scheme of a given order to a multi-level SDC scheme with comparable accuracy, the number of iterations are reduced (from 12 to 8 in our case). On the other hand, significant additional costs are introduced due to the additional f evaluations required by the coarse step and the transfer operations between fine and coarse quadrature nodes. However, the extra computational work of the multi-level SDC scheme can be distributed across multiple processors as demonstrated in the three remaining bars, which correspond to PFASST$(8,7)$ on two, four and eight processors. Hence, if the workload of PFASST$(8,7)$ is distributed across sufficiently many processors, then the total runtime becomes smaller than the time-to-solution of the serial SDC$(12,7)$ method. This is highlighted by

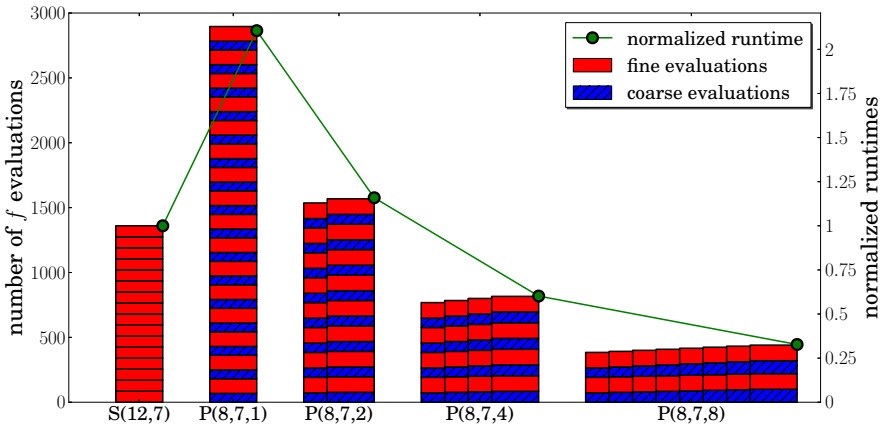


Fig. 2 Distribution of f evaluations on the coarse and fine level for SDC$(12,7)$ (abbreviated by S$(12,7)$) and PFASST$(8,7,Z)$ with $Z = 1,2,4,8$ time-processes (abbrev. by P$(8,7,Z)$). One block of coarse and fine evaluations corresponds to one time-step on one time-process (SDC is serial in time and evaluates only on the fine level). Depicted runtimes are normalized with respect to the SDC runtime (418 sec. for our test setup with with 10,000 particles, sixth-order algebraic kernel). All runs yield comparable accuracy for 16 time-steps.

the green line, which shows the runtime normalized by the runtime of SDC$(12,7)$. While PFASST$(8,7,2)$ is still slightly slower than SDC$(12,7)$, PFASST$(8,7,4)$ and PFASST$(8,7,8)$ show significant speedup. The cost of enlarging the problem, i. e. switching from SDC to a multi-level scheme, is compensated by the fact that this multi-level approach is amenable to parallelization while SDC itself is not.

4 Conclusion and Outlook

In this work, we have investigated the accuracy and convergence order of Spectral Deferred Correction (SDC) methods and their parallelization using the PFASST method. SDC provides a reliable, flexible, and generic mechanism to generate high-order and high-accuracy time integrators. We have shown that SDC and its time-parallel variant PFASST provide the theoretically expected convergence orders and accuracies on an example particle problem. In contrast to classical Runge-Kutta schemes, the convergence order and/or accuracy of SDC methods can easily be controlled by changing the number of iterations and/or quadrature points used, and the use of higher-order SDC methods allows much larger time-steps and hence fewer evaluations of the right-hand side. This is consistent with the increase of accuracy and also stability regions observed in [6].

Another key advantage of SDC methods is that that they can be parallelized in time with PFASST. Here, the careful union of fine and coarse SDC iterations leads to a high-order parallel-in-time integration scheme which relaxes Parareal's bound

on parallel efficiency and can provide significant speedup beyond space-only parallelization. In our test case, PFASST is more accurate for a given number of quadrature nodes and iterations, although for enough iterations both SDC and PFASST eventually provide the same solution. Moreover, we have demonstrated how the principle of "doing more to be faster" paves the way for temporal parallelism: the introduction of (possibly multi-level) coarsening in space and time increases the number of f evaluations significantly but also allows work to be distributed across many time-processors.

Here PFASST is used with temporal coarsening only, while considerably more parallel efficiency can be obtained by introducing both spatial and temporal coarsening. While grid-based spatial coarsening by multi-grid techniques is well understood, spatial coarsening of particles systems is less straightforward. One possibility is to control the quality of the approximation of f using multipole methods instead of direct summation [26]. Thus, the use of fast summation algorithms not only allows extreme-scale simulations as demonstrated in [27], but also introduces a promising way of particle-based spatial "coarsening".

Acknowledgements This research is partly funded by the Swiss "High Performance and High Productivity Computing" initiative HP2C; the Director, DOE Office of Science, Office of Advanced Scientific Computing Research, Office of Mathematics, Information, and Computational Sciences, Applied Mathematical Sciences Program, under contract DE-SC0004011; and the ExtreMe Matter Institute (EMMI) in the framework of the German Helmholtz Alliance HA216. Computing resources were provided by Jülich Supercomputing Centre under project JZAM04.

References

1. Barnes, J.E., Hut, P.: A hierarchical $\mathcal{O}(N \log N)$ force-calculation algorithm. Nature **324**(6096), 446–449 (1986)
2. Christlieb, A., Macdonald, C., Ong, B.: Parallel high-order integrators. SIAM Journal on Scientific Computing **32**(2), 818 (2010)
3. Christlieb, A., Ong, B., Qiu, J.: Comments on high order integrators embedded within integral deferred correction methods. Comm. Appl. Math. Comput. Sci **4**(1), 27–56 (2009)
4. Cottet, G.H., Koumoutsakos, P.: Vortex Methods: Theory and Applications, 2nd edn. Cambridge University Press (2000)
5. Cruz, F.A., Knepley, M.G., Barba, L.A.: PetFMM-A dynamically load-balancing parallel fast multipole library. International Journal for Numerical Methods in Engineering **79**(13), 1577–1604 (2010)
6. Dutt, A., Greengard, L., Rokhlin, V.: Spectral deferred correction methods for ordinary differential equations. BIT Numerical Mathematics **40**(2), 241–266 (2000)
7. Emmett, M., Minion, M.: Toward an efficient parallel in time method for partial differential equations. Comm. App. Math. and Comp. Sci. **7**(1), 105–132 (2012)
8. Gander, M.J., Vandewalle, S.: Analysis of the parareal time-parallel time-integration method. SIAM J. Sci. Comp. **29**(2), 556–578 (2007)
9. Gibbon, P., Speck, R., Karmakar, A., Arnold, L., Frings, W., Berberich, B., Reiter, D., Masek, M.: Progress in mesh-free plasma simulation with parallel tree codes. IEEE Transactions on Plasma Science **38**(9), 2367–2376 (2010). DOI 10.1109/tps.2010.2055165
10. Gibbon, P., Winkel, M., Arnold, L., Speck, R.: PEPC website (2012). URL `http://www.fz-juelich.de/ias/jsc/pepc`

11. Greengard, L., Rokhlin, V.: A fast algorithm for particle simulations. J. Comp. Phys. **73**(2), 325–348 (1987)
12. Huang, J., Jia, J., Minion, M.: Accelerating the convergence of spectral deferred correction methods. Journal of Computational Physics **214**(2), 633–656 (2006)
13. Jülich Supercomputing Centre: JUROPA/HPC-FF website (2012). URL `http://www.fz-juelich.de/jsc/juropa`
14. Layton, A.T., Minion, M.L.: Implications of the choice of quadrature nodes for picard integral deferred corrections methods for ordinary differential equations. BIT Numerical Mathematics **45**(2), 341–373 (2005)
15. Lions, J.L., Maday, Y., Turinici, G.: A "parareal" in time discretization of PDE's. C. R. Acad. Sci. – Ser. I – Math. **332**, 661–668 (2001)
16. Minion, M.L.: Semi-implicit spectral deferred correction methods for ordinary differential equations. Communications in Mathematical Sciences **1**(3), 471–500 (2003)
17. Minion, M.L.: A hybrid parareal spectral deferred corrections method. Comm. App. Math. and Comp. Sci. **5**(2), 265–301 (2010)
18. Nievergelt, J.: Parallel methods for integrating ordinary differential equations. Commun. ACM **7**(12), 731–733 (1964)
19. van Rees, W.M., Leonard, A., Pullin, D.I., Koumoutsakos, P.: A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. J. Comp. Phys. **230**, 2794–2805 (2011)
20. Ruprecht, D., Krause, R.: Explicit parallel-in-time integration of a linear acoustic-advection system. Computers & Fluids **59**, 72–83 (2012)
21. Salmon, J.K., Warren, M.S., Winckelmans, G.: Fast parallel tree codes for gravitational and fluid dynamical *N*-body problems. Int. J. Supercomp. App. **8**, 129–142 (1994)
22. Speck, R.: Generalized algebraic kernels and multipole expansions for massively parallel vortex particle methods. Ph.D. thesis, Universität Wuppertal (2011)
23. Speck, R., Arnold, L., Gibbon, P.: Towards a Petascale Tree Code: Scaling and Efficiency of the PEPC Library. J. Comp. Sci. **2**, 137–142 (2011)
24. Speck, R., Gibbon, P., Hofmann, M.: Efficiency and scalability of the parallel Barnes-Hut tree code PEPC. In: B. Chapman, F. Desprez, G.R. Joubert, A. Lichnewsky, F.J. Peters, T. Priol (eds.) Parallel Computing: From Multicores and GPU's to Petascale, *Adv. in Parallel Comp.*, vol. 19, pp. 35–42. IOS Press (2010)
25. Speck, R., Krause, R., Gibbon, P.: Parallel remeshing in tree codes for vortex particle methods. In: K.D. Bosschere, E.H. D'Hollander, G.R. Joubert, D. Padua, F. Peters, M. Sawyer (eds.) Applications, Tools and Techniques on the Road to Exascale Computing, *Advances in Parallel Computing*, vol. 22 (2012)
26. Speck, R., Ruprecht, D., Krause, R., Emmett, M., Minion, M., Winkel, M., Gibbon, P.: A massively space-time parallel N-body solver. In: Proceedings of the SC'12 International Conference for High Performance Computing, Networking, Storage and Analysis (2012). (Accepted)
27. Winkel, M., Speck, R., Hübner, H., Arnold, L., Krause, R., Gibbon, P.: A massively parallel, multi-disciplinary Barnes-Hut tree code for extreme-scale N-body simulations. Comp. Phys. Comm. **183**(4), 880–889 (2012)

Hybrid Space-Time Parallel Solution of Burgers' Equation

Rolf Krause[1] and Daniel Ruprecht[1,2]

1 Introduction

Many applications in high performance computing (HPC) involve the integration of time-dependent partial differential equations (PDEs). Parallelization in space by decomposing the computational domain is by now a standard technique to speed up computations. While this approach can provide good parallel scaling up to a large number of processors, it nevertheless saturates when the subdomains become too small and the time required for exchanging data starts dominating. Regarding the anticipated massive increase of available cores in future HPC systems, additional directions of parallelization are required to further reduce runtimes. This is especially important for time-critical applications like, for example, numerical weather prediction, where there exists a very strict constraint on the total time-to-solution for a forecast in order to be useful.

One possibility for providing such an additional direction of parallelization are parallel-in-time integration schemes. A popular scheme of this type is Parareal, introduced in [1, 7]. It has been applied successfully to a broad range of problems and also undergone thorough analytical investigation. A large number of corresponding references can be found, for example, in [6, 9].

While numerous works exist dealing with different aspects of Parareal in a purely time-parallel approach, there seem to be few studies that address the combination of Parareal with spatial parallelization, in particular with a focus on implementation. First results on combining Parareal with spatial domain decomposition are presented in [8]. While scaling of the algorithm is discussed, no runtimes are reported. In [12, 13], computing times for a pure MPI-based combination of Parareal with spatial domain decomposition for the two-dimensional Navier-Stokes equations are given, but with ambiguous results: Either a pure time-parallel or a pure space-parallel approach performed best, depending on the problem size. In [4], the capability of a purely MPI-based approach to speed up simulations for the 3D Navier-Stokes equations beyond the saturation of the spatial parallelization is shown. Extensive scaling tests for the "revisionist deferred correction" method (RIDC) for the linear heat equation, also in combination with domain decomposition, can be found in [3].

The present paper investigates the performance of a combination of a shared memory implementation of Parareal featuring explicit integrators with an MPI-based parallelization of a stencil-based spatial discretization into a hybrid (see [10])

[1]Institute of Computational Science, Università della Svizzera italiana, Via Giuseppe Buffi 13, 6900 Lugano, Switzerland, {rolf.krause,daniel.ruprecht}@usi.ch .[2] Mathematisches Institut, Heinrich-Heine-Universität, Universitätsstrasse 1, 40225 Düsseldorf, Germany

space-time parallel method. The code is an extension of the purely time-parallel, OpenMP-based implementation used in [11]. Using shared memory for Parareal avoids communication of volume data by message passing and thus reduces the memory footprint of the code.

2 Algorithm and Implementation

The starting point for Parareal is an initial value problem

$$\frac{d\mathbf{q}}{dt} = \mathbf{f}(\mathbf{q}),\ \mathbf{q}(0) = \mathbf{q}_0 \in \mathbb{R}^d, \tag{1}$$

where in the present work, the right hand side $\mathbf{f}$ stems from the spatial finite difference discretization of some PDE on a rectangular domain $\Omega \subset \mathbb{R}^2$. The spatial parallelization uses a standard non-overlapping decomposition of the domain, allowing for a distributed computation of $\mathbf{f}(\mathbf{q})$, where every MPI-process handles the degrees-of-freedom of one subdomain and ghost-cell values are exchanged at the boundaries. The implementation described below can be used for all integrators that involve only straightforward evaluations of $\mathbf{f}$, that is explicit methods or implicit methods where the arising linear or nonlinear system is solved with e.g. a fixed point iteration. For more complex solvers, e.g. a multi-grid method, a hybrid strategy will be more involved, because other parts like restriction or interpolation would have to be included in the hybrid paradigm as well.

2.1 Parareal

Parareal allows one to parallelize the integration of (1) by combining a number of time-steps into one coarse time-slice and performing an iteration where multiple time-slices are treated concurrently. Let $\mathscr{F}_{\delta t}$ denote a numerical integration scheme of suitable accuracy, using a time-step δt. A second integration scheme is required, typically called $\mathscr{G}_{\Delta t}$, using a time-step $\Delta t \gg \delta t$, which has to be much cheaper in terms of computation time but can also be much less accurate. Denote by

$$\tilde{\mathbf{q}}_{\mathrm{g}} = \mathscr{G}_{\Delta t}(\mathbf{q}, \tilde{t}, t), \quad \tilde{\mathbf{q}}_{\mathrm{f}} = \mathscr{F}_{\delta t}(\mathbf{q}, \tilde{t}, t) \tag{2}$$

the result of integrating forward in time from an initial value $\mathbf{q}$ at time t to a time $\tilde{t} > t$ using $\mathscr{G}_{\Delta t}$ or $\mathscr{F}_{\delta t}$. Parareal uses $\mathscr{G}_{\Delta t}$ to produce approximate solutions at nodes $(t_i)_{i=0,\ldots,N_{\mathrm{c}}}$ of a coarse temporal mesh (lines 2 – 4 in Algorithm 1). These guesses are then used as initial values for running $\mathscr{F}_{\delta t}$ concurrently on all N_{c} time intervals $[t_i, t_{i+1}]$ (lines 6–10). A correction is then propagated sequentially by another sweep of $\mathscr{G}_{\Delta t}$ (lines 11 – 13). The procedure is iterated and converges towards the solution that would be obtained by running $\mathscr{F}_{\delta t}$ sequentially from t_0 to $t_{N_{\mathrm{c}}}$. For a detailed

Algorithm 1 Parareal algorithm implemented with OpenMP using N_c threads

1: $\mathbf{q}_0^0 = \mathbf{q}_0$, $k := 0$
2: **for** $i = 0$ to $N_\mathrm{c} - 1$ **do**
3: $\quad \mathbf{q}_{i+1}^0 = \mathscr{G}_{\Delta t}(\mathbf{q}_i^0, t_{i+1}, t_i)$
4: **end for**
5: **repeat**
6: $\quad$ omp parallel for
7: $\quad$ **for** $i = 0$ to $N_\mathrm{c} - 1$ **do**
8: $\quad\quad \tilde{\mathbf{q}}_{i+1}^k = \mathscr{F}_{\delta t}(\mathbf{q}_i^k, t_{i+1}, t_i)$
9: $\quad$ **end for**
10: $\quad$ omp end parallel for
11: $\quad$ **for** $i = 0$ to $N_\mathrm{c} - 1$ **do**
12: $\quad\quad \mathbf{q}_{i+1}^{k+1} = \mathscr{G}_{\Delta t}(\mathbf{q}_i^{k+1}, t_{i+1}, t_i) + \tilde{\mathbf{q}}_{i+1}^k - \mathscr{G}_{\Delta t}(\mathbf{q}_i^k, t_{i+1}, t_i)$
13: $\quad$ **end for**
14: $\quad k := k + 1$
15: **until** $k = N_\mathrm{it}$

explanation and properties of the algorithm we refer to [6] and references therein. Note that an MPI-based implementation of Parareal requires communication of full volume data in line 12, which is avoided by the shared memory parallelization in time used here.

For a given time interval $[t_0, t_{N_\mathrm{c}}]$, denote by N_f the number of fine steps required to integrate from $t = t_0$ to $t = t_{N_\mathrm{c}}$, by τ_c and τ_f the execution time of one single coarse or fine time-step and by N_it the number of performed iterations. Further, assume that $\mathscr{G}_{\Delta t}$ always performs one single step, so that N_c is also the number of coarse steps between t_0 and t_{N_c}. The speedup obtainable by Parareal for a given number of processors can be estimated by

$$s(N_p) \approx \frac{1}{(1+N_\mathrm{it})\,\frac{N_\mathrm{c}}{N_\mathrm{f}}\,\frac{\tau_\mathrm{c}}{\tau_\mathrm{f}} + \frac{N_\mathrm{it}}{N_\mathrm{p}}} \leq \frac{N_\mathrm{p}}{N_\mathrm{it}}. \tag{3}$$

Note that the maximum parallel efficiency is bounded by $1/N_\mathrm{it}$. Because of this limit, Parareal is commonly considered on top of a saturated spatial parallelization for problems where minimizing time-to-solution is critically important. Recently, a new scheme named PFASST, based on a combination of Parareal with spectral deferred correction methods, has been introduced in [5].

2.2 Implementation

For the OpenMP-based parallelization sketched in Algorithm 1 to be efficient, the implementation of the fine propagator $\mathscr{F}_{\delta t}$ has to be suitably designed for multithreading. This involves a number of technical issues like taking care of "non-uniform memory access" (NUMA) inside compute nodes by ensuring that the data

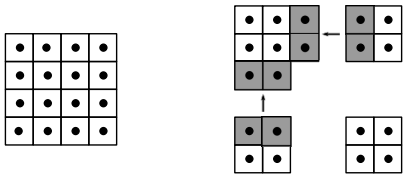


Fig. 1 Sketch of a decomposition of a 4x4 cells domain (left) into 4 sub-domains with 2x2 cells each (right). Cell-centers are marked as dots. The grey cells mark the halo values that have to be send to the processor handling the upper left sub-domain before each evaluation of **f** if a simple 5-point star is used. For stencils with wider support, the halos also need to be wider and communication between diagonally adjacent processors might be required. In the time-parallel OpenMp version, halo data has to be exchanged for each thread. In the implementation used here, the master thread handles all halo exchanges as sketched in Figure 2.

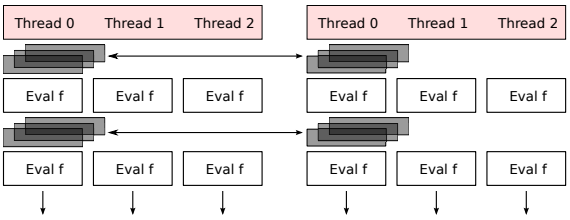


Fig. 2 Flow chart of halo exchange in funneled mode with 2 nodes, each running 3 threads: Before each evaluation of the right hand side **f**, the master thread (thread 0) exchanges up-to-date halo values (represented by three grey bars) for all threads with the other node. The other threads are idle during communication. After communication has finished, all threads continue with evaluating **f**. Synchronization is achieved by the OpenMp BARRIER directive while MPI calls are enclosed in MASTER directives to ensure they are only executed by the master thread.

a core accesses while running a thread is located in the memory closest to this core. A detailed introduction into efficient OpenMp programming can be found in [2].

2.2.1 Ghost-Cell Exchange

To combine the OpenMp implementation of Parareal with parallelization in space, frequent exchange of boundary values between processors handling different sub-domains is necessary: Figure 1 sketches the decomposition of a 4×4 cell domain into 4 sub-domains. In order to evaluate e.g. a standard five-point stencil discrete Laplacian, every processor needs to receive a ”halo” of up-to-date values before evaluating the stencil (halo cells for the upper left sub-domain are marked in grey in Figure 1). Communication of these halo data is done here through message passing using MPI.

For using MPI in conjunction with OpenMp, different options exists for the initialization of the MPI library that govern how MPI routines can be called by different threads. Here, we use the option MPI_THREAD_FUNNELED which allows only the master thread to make calls to the MPI library. As the ghost-cell communication

in $\mathscr{F}_{\delta t}$ takes place in the multithreaded part of the code, suitable OpenMP directives have to be used to synchronize threads and ensure compliance with the funneled option (OMP_BARRIER and OMP_MASTER). The coarse integrator is outside the parallel OpenMp region in Algorithm 1 so that no thread synchronization is required there. Organization of the ghost-cell exchange is sketched in Figure 2: Prior to every evaluation of the right hand side function **f**, the master thread (thread 0) exchanges halo data for all threads on the node. While the master thread is busy communicating, the other threads are idle. This "idle threads problem" is one of the drawbacks of the funneled approach pointed out in [10]. Then, after the master thread has finished communicating, all threads continue with the computation of **f** and update the solution according to the integration method used for $\mathscr{F}_{\delta t}$. After every update (in case of a Runge-Kutta method for example, that means after every stage), the new halo values have to be exchanged again by the master thread before the next evaluation of **f** and so on.

3 Numerical Results

The performance of the hybrid space-time parallel approach is analyzed here for the two-dimensional, nonlinear, viscous Burgers equation

$$u_t + uu_x + uu_y = \nu \Delta u \tag{4}$$

on a domain $[-2,2] \times [-2,2]$ with initial value

$$u(x,y,0) = \sin(2\pi x)\sin(2\pi y), \tag{5}$$

a parameter $\nu = 0.02$, a mesh width $\Delta x = \Delta y = 1/40$ and periodic boundary conditions. A two-dimensional decomposition of the domain into square or rectangular subdomains, depending on the number of MPI-processes, is performed and a cartesian communicator for MPI is used. Parareal uses time-steps $\Delta t = 2 \times 10^{-3}$ and $\delta t = 2 \times 10^{-5}$. For $\mathscr{G}_{\Delta t}$, the spatial discretization uses 3rd-order upwind finite differences for the advection term and 2nd-order centered differences for the Laplacian, while $\mathscr{F}_{\delta t}$ uses a 5th-order upwind stencil for the advection and a 4th-order centered stencil for the Laplacian. Hence, a two-cell wide halo has to be exchanged in the coarse and a three-cell wide halo in the fine propagator. The simulations are run until $T = 0.5$ and $\mathscr{G}_{\Delta t}$ always performs one single step per coarse interval, so the number of restarts of Parareal depends on the number of threads assigned for the temporal parallelization. A forward Euler scheme is used for $\mathscr{G}_{\Delta t}$ and a Runge-Kutta-2 scheme for $\mathscr{F}_{\delta t}$. To assess accuracy, a reference solution with $\delta t/10$ is computed sequentially. With a fixed number of $N_{\text{it}} = 3$ in Parareal, the relative $\|\cdot\|_\infty$-error of the time-parallel solution is $\varepsilon_{\text{para}} \approx 2.2 \times 10^{-8}$ and of the time-serial solution $\varepsilon_{\text{seq}} \approx 1.8 \times 10^{-8}$, so that both solutions are of comparable accuracy. The coarse integrator run alone results in $\varepsilon_{\text{coarse}} \approx 2.9 \times 10^{-2}$.

# MPI-P	time-serial	hybrid Parareal	speedup	# MPI-P	time-serial	hybrid Parareal	speedup
1	59.9 s	29.5 s	2.0	1	16.4 s	7.3 s	2.2
2	34.6 s	15.4 s	2.2	2	10.5 s	4.9 s	2.1
4	21.2 s	9.4 s	2.3	4	6.9 s	3.3 s	2.1
8	14.2 s	6.0 s	2.4	8	4.7 s	2.2 s	2.1
16	9.2 s	4.2 s	2.2	16	3.3 s	1.5 s	2.2
20	9.5 s	–	–	20	4.5 s	–	–

Table 1 Runtimes of the time-serial code and the hybrid Parareal code using 8 threads on each node for different numbers of spatial sub-domains, each corresponding to one MPI process. Shown are runtimes for a grid with 160×160 cells (left) and for a grid with 80×80 cells (right). Note that using more than 16 sub-domains no longer reduces runtime of the serial code in both cases.

The used machine is a cluster consisting of 42 nodes, each containing 2 quad-core AMD Opteron CPUs with 2,700 MHz and 16 GB RAM per node. In the example below, the time parallelization always uses eight threads per node, in order to utilize one full node. The nodes are connected by an INFINIBAND network.

3.1 Runtimes and Scaling

Reported runtimes are measured with the MPI_WTIME routine provided by MPI and do not contain I/O operations.

3.1.1 Speedup from Parareal

With the used parameters, the speedup obtainable by Parareal using eight threads is bounded by $s \leq 2.57$ according to (3). The ratio $\tau_c/\tau_f = 0.35$ has been determined experimentally fby running $\mathscr{G}_{\Delta t}$ and $\mathscr{F}_{\delta t}$ serially on a single core. The value varies when using multiple processes, but the effects of the variation on the speedup estimate are small. Table 1 (left) shows the runtimes of the time-serial and the hybrid Parareal solution for different numbers of subdomains and corresponding MPI-processes. To further illustrate performance of the approach, runtimes for the 80×80 cell mesh are also shown (right). Runtimes obtained for a 40×40 mesh not shown here indicate similar speedups from Parareal using one, two and four MPI-processes as well as no further reduction of runtime of the time-serial code if more than four MPI-processes are used.

While the time-serial solution assigns each process to one core, the time-parallel solution assigns each process to one node and uses the eight cores inside the node for the temporal parallelization. In both cases, the speedups from Parareal actually achieved by the hybrid implementation are between 78% and 93% of the theoretical maximum, despite the overhead caused by the funneled mode, supporting the efficiency of the hybrid space-time parallel approach.

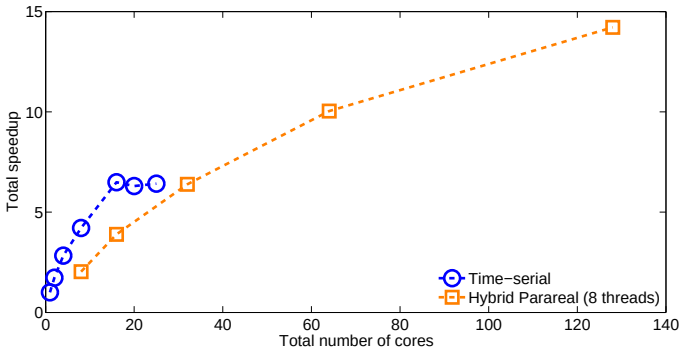


Fig. 3 Total speedup achieved by the space-parallel, time-serial (blue) and the hybrid space-time-parallel scheme (red) depending on the total number of used cores for the 160×160 cell mesh.

3.1.2 Total scaling

As discussed above, one essential motivation for time-parallel schemes is to provide an additional direction of parallelization to achieve further reduction of time-to-solution after spatial parallelization saturates. Figure 3 shows the total speedup, that is compared against the time-serial solution on one core, for the time-serial and hybrid Parareal scheme. Because the considered problem is quite small and the underlying stencil-based discretization is comparably cheap to evaluate in terms of computation time, the pure spatial parallelization scales only to 16 cores (cf. Table 1). Beyond that point, using more cores does not further reduce runtime. Also, near perfect scaling is seen only up to two cores, after this the parallel efficiency is noticeable less than one. Note that the slow increase in speedup for the hybrid scheme is caused by the efficiency bound (3) of Parareal: For lower numbers of cores where the spatial parallelization is not yet saturated, the time-serial version performs better, because the efficiency of the parallelization in space, although no longer optimal, is still better than that of the time-parallel scheme. The advantage of the space-time-parallel scheme is that it can provide a significantly greater overall speedup. Hence, for a time-critical application where minimizing time-to-solution is of paramount importance and a purely spatial parallelization does not provide sufficient runtime reduction, a space-time parallel scheme can reduce runtime below some critical threshold if sufficient computational resources are available. The example clearly demonstrates the potential of the hybrid space-time parallelization to provide runtime reductions beyond the saturation of the space parallelization.

4 Conclusions

A shared memory implementation of the Parareal parallel-in-time integration scheme is combined with a standard distributed memory parallelization of a stencil-based

spatial discretization. In the resulting hybrid space-time parallel scheme, each spatial subdomain is handled by one MPI-process which is assigned to one compute node. The time-slices from Parareal are assigned to different threads spawned by the process, with each thread running on one core of the node. The capability of the hybrid implementation to provide runtime reduction beyond the saturation of the spatial parallelization is documented.

Acknowledgements This work is funded by the Swiss "High Performance and High Productivity Computing" initiative HP2C.

References

1. Bal, G., Maday, Y.: A "parareal" time discretization for non-linear PDE's with application to the pricing of an american put. In: L. Pavarino, A. Toselli (eds.) Recent Developments in Domain Decomp. Meth., *LNCSE*, vol. 23, pp. 189–202. Springer Berlin (2002)
2. Chapman, B., Jost, G., van der Pas, R.: Using OpenMp: Portable shared memory parallel programming. Scientific and Engineering Computation Series. The MIT press, Cambridge, London (2008)
3. Christlieb, A.J., Haynes, R., Ong, B.W.: A parallel space-time algorithm. SIAM Journal on Scientific Computing **34**, C233–C248 (2012)
4. Croce, R., Ruprecht, D., Krause, R.: Parallel-in-space-and-time simulation of the three-dimensional, unsteady Navier-Stokes equations for incompressible flow. ICS-Preprint 2012-03 (2012)
5. Emmett, M., Minion, M.L.: Toward an efficient parallel in time method for partial differential equations. Comm. App. Math. and Comp. Sci. **7**, 105–132 (2012)
6. Gander, M.J., Vandewalle, S.: Analysis of the parareal time-parallel time-integration method. SIAM J. Sci. Comp. **29**(2), 556–578 (2007)
7. Lions, J.L., Maday, Y., Turinici, G.: A "parareal" in time discretization of PDE's. C. R. Acad. Sci. – Ser. I – Math. **332**, 661–668 (2001)
8. Maday, Y., Turinici, G.: The parareal in time iterative solver: A further direction to parallel implementation. In: R. Kornhuber, et al. (eds.) Domain Decomposition Methods in Science and Engineering, *LNCSE*, vol. 40, pp. 441–448. Springer, Berlin (2005)
9. Minion, M.L.: A hybrid parareal spectral deferred corrections method. Comm. App. Math. and Comp. Sci. **5**(2), 265–301 (2010)
10. Rabenseifner, R., Hager, G., Jost, G.: Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes. In: 17th Euromicro International Conference on Parallel, Distributed and Network-based processing, pp. 427–436 (2009)
11. Ruprecht, D., Krause, R.: Explicit parallel-in-time integration of a linear acoustic-advection system. Computers & Fluids **59**, 72–83 (2012)
12. Trindade, J.M.F., Pereira, J.C.F.: Parallel-in-time simulation of the unsteady Navier-Stokes equations for incompressible flow. Int J. Numer. Meth. Fluids **45**, 1123–1136 (2004)
13. Trindade, J.M.F., Pereira, J.C.F.: Parallel-in-time simulation of two-dimensional, unsteady, incompressible laminar flows. Num. Heat Trans., Part B **50**, 25–40 (2006)

Optimized interface preconditioners for the FETI method

Martin J. Gander[1] and Hui Zhang[1]

1 Motivation

In the past two decades, the FETI method introduced in [10] and its variants have become a class of popular methods for the parallel solution of large-scale finite element problems, see e.g. [11], [9], [14], [15], [8]. A key ingredient in this class of methods is a good preconditioner for the dual Schur complement system whose operator is a weighted sum of subdomain Neumann to Dirichlet (*NtD*) maps. One choice is the so-called Dirichlet preconditioner, which is the primal Schur complement, i.e. a weighted sum of subdomain Dirichlet to Neumann (*DtN*) maps. The Dirichlet preconditioner is quasi-optimal in the sense that together with an appropriate coarse space, it leads to a polylogarithmic condition number in H/h, see e.g. [14]. However, in terms of total CPU time, often a cheaper alternative called the lumped preconditioner performs better [11, 8].

We show here that the lumped preconditioner can be further improved by introducing parameters into the tangential interface operator and optimizing them to get condition numbers as small as possible while keeping the cost of the preconditioner low. Since these preconditioners, like the lumped preconditioner, only involve computations along the interface, and no computations in the interior of subdomains, we call them *interface preconditioners*.

We consider the model problem

$$\begin{cases} -u_{xx} - u_{yy} = f, & (x,y) \in \mathbb{R}^2 \\ \lim_{(x,y)\to\infty} u = 0, \end{cases}$$

which can be decomposed into two non-overlapping subproblems as follows:

$$\begin{cases} -(u_1)_{xx} - (u_1)_{yy} = f, & (x,y) \in (-\infty,0) \times \mathbb{R}, \\ (u_1)_x = \lambda, & (x,y) \in \{0\} \times \mathbb{R}, \\ \lim_{(x,y)\to\infty} u_1 = 0, \end{cases} \tag{1}$$

$$\begin{cases} -(u_2)_{xx} - (u_2)_{yy} = f, & (x,y) \in (0,\infty) \times \mathbb{R}, \\ (u_2)_x = \lambda, & (x,y) \in \{0\} \times \mathbb{R}, \\ \lim_{(x,y)\to\infty} u_2 = 0, \end{cases} \tag{2}$$

$$\frac{1}{2}u_1 - \frac{1}{2}u_2 = 0, \quad (x,y) \in \{0\} \times \mathbb{R}. \tag{3}$$

[1]University of Geneva, 2-4 rue du Liévre, Case postale 64, e-mail: {martin.gander}{hui.zhang}@unige.ch

The FETI method takes the common Neumann trace λ as unknown and the equation to be solved for λ is defined by (3). To analyze the operator of the equation for λ, we let $f = 0$ and do a Fourier transform in $y \in \mathbb{R}$ for (1), (2) and (3) to obtain

$$\begin{cases} -(\hat{u}_1)_{xx} - k^2\hat{u}_1 = 0, & x \in (-\infty, 0), \\ (\hat{u}_1)_x = \hat{\lambda}, & x = 0, \\ \lim_{x\to-\infty} \hat{u}_1(x,k) = 0, \end{cases} \tag{4}$$

$$\begin{cases} -(\hat{u}_2)_{xx} - k^2\hat{u}_2 = 0, & x \in (0, \infty), \\ (\hat{u}_2)_x = \hat{\lambda}, & x = 0, \\ \lim_{x\to\infty} \hat{u}_2(x,k) = 0, \end{cases} \tag{5}$$

and

$$\frac{1}{2}\hat{u}_1 - \frac{1}{2}\hat{u}_2 = 0, \quad x = 0. \tag{6}$$

The subdomain solutions $\hat{u}_i, i = 1,2$ can be obtained from (4) and (5), and substituting them into the left hand side of (6) yields the equation for $\hat{\lambda}$,

$$\hat{F}\hat{\lambda} := \frac{1}{\sqrt{k^2}}\hat{\lambda},$$

where $\hat{F}$ is the symbol of the averaged NtD operator F. Similarly, one can obtain the symbol of the Dirichlet preconditioner (i.e. the averaged DtN operator): it is exactly $\hat{F}^{-1}$, which means that the Dirichlet preconditioner is an exact preconditioner for our symmetric partition into two subdomains. However, using the Dirichlet preconditioner requires to solve the Dirichlet boundary value problems on the subdomains, which are in addition to the Neumann boundary value problems involved in F.

As a cheaper alternative, Farhat and Roux introduced the lumped preconditioner for F, see e.g. [11], which corresponds to the submatrix of the assembled matrix for the original problem restricted to the interface. Here we explain it as an operator at the continuous level, that is $P_L^{-1} := -\partial_{yy} + p$ acting on the interface, where $p = O(h^{-2})$. To see this, let us consider a 5-point stencil discretization of the minus Laplacian (see the left part of the following illustration)

$$\frac{1}{h^2}\begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}, \qquad \frac{1}{h^2}\begin{bmatrix} -1 \\ 4 \\ -1 \end{bmatrix}.$$

Assuming the interface is along the vertical direction, the lumped preconditioner corresponds to a 3-point stencil along interface, shown on the right part of the above illustration. So the symbol of the preconditioned operator will be

$$\hat{P}_L^{-1}\hat{F} := \frac{k^2 + p}{\sqrt{k^2}}.$$

Note that practically $|k|$ varies in $[k_{\min}, k_{\max}] := [\frac{\pi}{H}, \frac{\pi}{h}]$, where H is the domain size and h is the mesh size, both along the y direction, and that usually we have $\sqrt{p} \in$

$[k_{\min}, k_{\max}]$. In this case, the spectra of $\hat{P}_L^{-1}\hat{F}$ are bounded by

$$\sigma(\hat{P}_L^{-1}\hat{F}) \subset [2\sqrt{p}, \max\{k_{\min} + \frac{p}{k_{\min}}, k_{\max} + \frac{p}{k_{\max}}\}].$$

If we fix H and let $h \to 0$, we will find that the condition number of $P_L^{-1}F$ is $O(h^{-1})$. So the drawback of the lumped preconditioner is that the condition number deteriorates at the same rate as the unpreconditioned method as the mesh size tends to zero. This is conforming to the result in [9] where it was also pointed out that the lumped preconditioner has a favorable spectral distribution for the Conjugate Gradient method. However, in our special cases of numerical experiments, we have not found this superiority, see Sec. 3 (in which we did use the original form of the lumped preconditioner).

Since the symbol of F is $\hat{F} = \frac{1}{\sqrt{k^2}}$, it is clear that an exact preconditioner for F is the square-root of the Laplacian operator on the interface. We already know that the Dirichlet preconditioner implements the square-root through subdomain solves. There are also other ways to approximate the square-root or its inverse, the latter is useful for the primal Schur complement methods. Some are based on the idea of FFT and its extensions, see e.g. [7, 4, 13]. Two multilevel methods are proposed in [5]. In [16], the Green's function is used for approximating the inverse square-root in general geometry. In the more recent approach [3, 2], a Krylov subspace method is adopted for the approximate application of the inverse square-root. In the context of integral equation methods for scattering problems, Padé approximation is adopted for preconditioning, see e.g. [6]. Our work is more related to that of [1][1], in which the ideas of using quadratic approximations and minimizing the condition number were first presented. The first difference of our work from that of [1] lies in the problems studied: the positive definite Helmholtz equation is considered in [1] while we study the Poisson equation here. The second difference is that we propose two new approaches, in addition to the quadratic approximation.

2 Optimized Interface Preconditioners

In this section we will introduce some approximations of the square-root of the interface Laplacian, which define our new preconditioners. Parameters involved in these approximations will be optimized so that condition numbers of the corresponding preconditioned operators are as small as possible.

We first consider the preconditioner whose symbol is of the form $\hat{P}^{-1} := k^2 + p$, the same as that of the lumped preconditioner. We now optimize however the parameter $p \geq 0$ by solving the minimization problem

$$\min_{p\geq 0} \mathrm{cond}(P^{-1}F) = \min_{p\geq 0} \frac{\max_{k\in[k_{\min},k_{\max}]}(k^2+p)/\sqrt{k^2}}{\min_{k\in[k_{\min},k_{\max}]}(k^2+p)/\sqrt{k^2}}. \quad (7)$$

[1] We only discovered this reference when we already finished our present investigation.

Theorem 1. *The solution of problem (7) is given by* $p^* = k_{\min}k_{\max}$. *In particular, if* $k_{\min} = O(1)$ *and* $k_{\max} = O(h^{-1})$, *we have* $\mathrm{cond}(P^{-1}F) = O(h^{-1/2})$ *when* $p = p^*$.

Remark 1. It is also possible to include the first-order derivative into the preconditioner. But in that case, symmetry is destroyed, and minimizing the condition number is then not necessarily the relevant goal.

In the second approach, the symbol of the preconditioner is chosen to be of the form

$$\hat{P}^{-1} = \frac{p_0 + p_2k^2 + k^4}{q + k^2}, \tag{8}$$

and we optimize p_0, p_2, q by solving the minimization problem

$$\min_{p_0,p_2,q\geq 0} \mathrm{cond}(P^{-1}F) = \min_{p_0,p_2,q\geq 0} \frac{\max_{k\in[k_{\min},k_{\max}]}\rho(k)}{\min_{k\in[k_{\min},k_{\max}]}\rho(k)},$$

where $\rho(k)$ is the symbol of the preconditioned operator, i.e.

$$\rho(k) := \frac{p_0 + p_2k^2 + k^4}{(q + k^2)\sqrt{k^2}}.$$

Theorem 2. *Assume* $k_{\max} = Ch^{-1}$ *and let* $p_0 = p_2 = k_{\max}^{4/3}\left(2k_{\min} + \frac{2}{k_{\min}}\right)^{2/3}$, $q = \left(k_{\min} + \frac{1}{k_{\min}}\right)^{4/3}(k_{\max}/2)^{2/3}$. *Then we have* $\mathrm{cond}(P^{-1}F) = O(h^{-1/3})$.

Remark 2. We found numerically that the smaller $k_{\min}$ is, the smaller h needs to be before the asymptotics set in. We also observed that there exist better choices of parameters in the pre-asymptotic regime, but a formula still needs to be found.

Remark 3. There are many possible ways to implement (8) in the physical domain. We found that a good way in practice is formally given by

$$P^{-1} = (-\partial_{yy} + r_0)(-\partial_{yy} + q)^{-1}(-\partial_{yy} + r_2),$$

where r_0, r_2 are related to p_0, p_2 of (8) such that $k^4 + p_2k^2 + p_0 = (k^2 + r_0)(k^2 + r_2)$.

Now we propose a third approach for approximating the square root. Suppose we have a good preconditioner $\tilde{A}$ (e.g. Jacobi) for an operator A such that

(i) $\|S\| < 1$ where $S := I - \tilde{A}^{-1}A$ is the iteration operator,
(ii) $\tilde{A}^{1/2}$ is cheap to apply,
(iii) A commutes with $\tilde{A}$ (this can be omitted in practice).

Then, using Newton's binomial series, we have

$$A^{1/2} = \tilde{A}^{1/2}(I - S)^{1/2} = \tilde{A}^{1/2}\sum_{i=0}^{\infty}\binom{\frac{1}{2}}{i}(-S)^i.$$

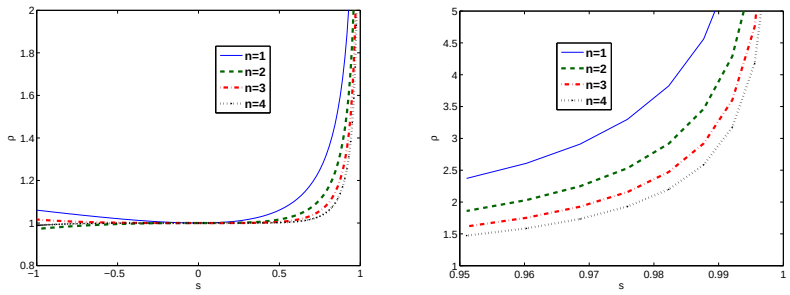


Fig. 1 Values of the symbol (10) with p_i the binomial coefficient in (9).

A preconditioner for $A^{-1/2}$ can be obtained by truncating the infinite series,

$$P^{-1} = \tilde{A}^{1/2} \sum_{i=0}^{n} \binom{\frac{1}{2}}{i} (-S)^i, \tag{9}$$

so n iterations of S are needed in one application of P^{-1}. We can also consider the more general polynomial

$$P^{-1} = \tilde{A}^{1/2} \sum_{i=0}^{n} p_i (-S)^i.$$

The right preconditioned operator is then

$$A^{-1/2}P^{-1} = T \sum_{i=0}^{n} p_i (-S)^i, \quad T := A^{-1/2}\tilde{A}^{1/2}.$$

Assume the symbol of S to be $s \in [s_{\min}, s_{\max}] \subset (-1,1)$ and the symbol of T to be $\hat{T} = \frac{1}{\sqrt{1-s}}$, which can be obtained for example by Fourier analysis. Hence, the symbol of the preconditioned operator is

$$\rho(s) := \hat{A}^{-1/2}\hat{P}^{-1} = \frac{1}{\sqrt{1-s}} \sum_{i=0}^{n} p_i (-s)^i. \tag{10}$$

In the case of truncation of the binomial series and $n = 1,\ldots,4$, the symbols are plotted in Fig.1. This clearly shows that the symbol value tends to infinity as s goes to one. In fact, we have $\rho = O(t^{-1/2})$ for $t := 1 - s \to 0$.

For example, when S is Jacobi for the 1d discrete Laplacian, we have $s \in \{\cos(\frac{j\pi}{N}),\ j = 1,\ldots,N-1\}$, where the mesh size is $h = 1/N$ and we assumed Dirichlet conditions on the boundary. In this case, we have $\max_{[-1,s_{\max}]} \rho(s) = O(h^{-1})$ no matter how many orders are kept in the truncation! To make things worse, the discrete points in s are more clustered near $s = \pm 1$ than elsewhere.

Remark 4. Since the direct truncation of the binomial series is really good away from low frequencies (small j), it is natural to approximate the low frequency part by a coarse grid or multigrid. We will however not investigate this further here.

The idea to improve is optimizing the parameters $\{p_i\}$ such that the corresponding condition number is minimized. We begin with the approximation of order $n = 1$.

Theorem 3. *Let $s \in [-1, s_{\max}]$ with $0 < s_{\max} < 1$, $n = 1$ and assume $p_1 = 1$ is used for the preconditioned operator (10). If the operator is positive definite, then the condition number of the preconditioned operator is minimized if and only if*

$$p_0 = \frac{2s_{\max} + \sqrt{2 - 2s_{\max}}}{2 - \sqrt{2 - 2s_{\max}}},$$

in which case $\mathrm{cond}(P^{-1}A^{-1/2}) = O(t^{-1/4})$ *as* $t := 1 - s_{\max} \to 0$.

We also tried the approximation of order $n = 2$. The scaling of the condition number when s goes to one is *not improved for the exponent* but is improved for the constant. We do however not have closed formulas for the optimized parameters when $n \geq 2$.

3 Numerical experiments

All the numerical experiments are coded in FreeFem++ [12] with P1 elements. We solve homogeneous equations on $\Omega = (0,1)^2$ with the zero solution. We take a random initial guess for the CG iterations which stop when the relative preconditioned residual norms are less than 10^{-15}. It is worthwhile to note that the proposed preconditioners involve only *integer-order* differential operators easily implementable as matrices from standard discretization (FFT is unnecessary). So in methodology, they are applicable to general geometry though the optimal parameters could change.

First, we solve the Laplace equation in two equal subdomains. The maximum errors of the iterates to the exact zero solution are illustrated in Fig. 2 against the iteration numbers, from which we can see that with the optimized interface preconditioners the iterations converge faster than without or with the lumped preconditioner, and the optimized rational preconditioners eventually outperform the others in terms of iteration numbers as the mesh size h becomes small.

Next, we consider a diffusion problem with smoothly varying coefficient,

$$\begin{aligned} -\nabla \cdot (a(x,y)\nabla u) &= 0, \quad (x,y) \in (0,1)^2, \\ u &= 0, \quad \text{if } xy(1-x)(1-y) = 0, \end{aligned} \tag{11}$$

where $a = \nu x^2 y^2 + 0.1$. The coefficient a is continuous but varies along the interface. To study the effect of the variation, we take the constant ν to be 1, 10, 100, and 1000. We use a fifth-order quadrature rule to ensure accurate numerical integration in the discretization. The results using two subdomains are shown in Fig. 3, and clearly show the robustness of the optimized interface preconditioners, except for the one based on the Jacobi preconditioner.

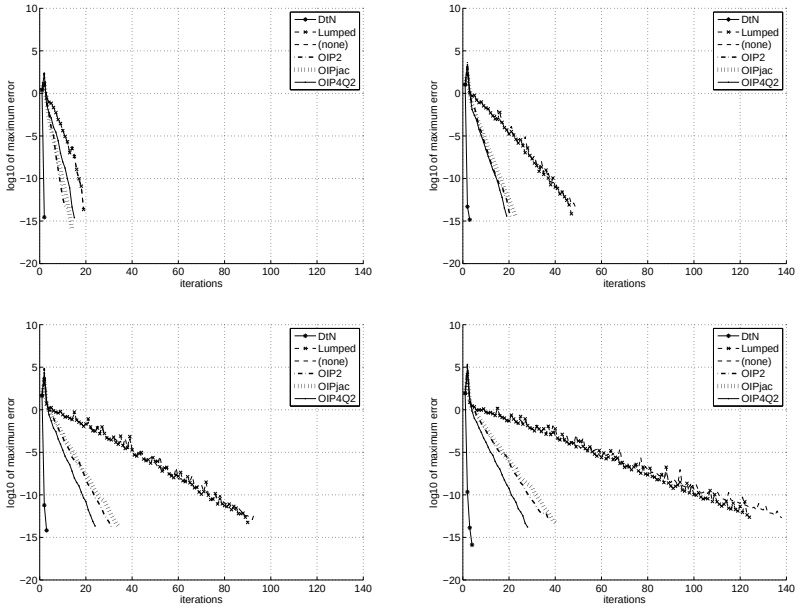


Fig. 2 Maximum errors between iterates and the FEM solutions for the Laplacian in the unit square for $h = 1/16, 1/64, 1/256, 1/512$.

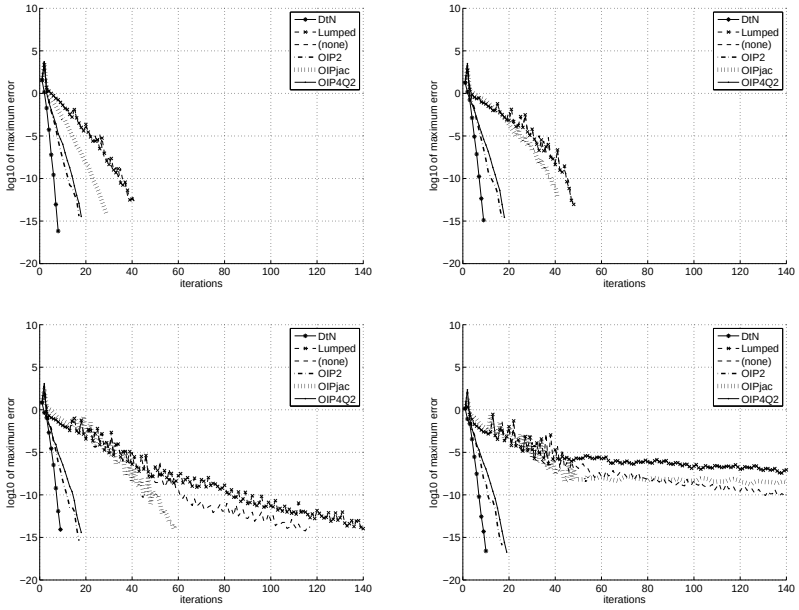


Fig. 3 Maximum errors between iterates and the FEM solutions for (11) with $h = 1/32$ for $\nu = 1, 10, 100, 1000$.

Remark 5. For the quadratic and the quartic/quadratic approximation, we adapted the interface Laplacian to $\partial_y(a(x,y)\partial_y)$ and at the same time multiplied the optimized parameters with $a(x,y)$. For the Jacobi induced preconditioner, we still use the interface Laplacian operator, which is better than using the diffusion operator.

Acknowledgements The work is supported by University of Geneva. The second author is also partially supported by the International Science and Technology Cooperation Program of China (2010DFA14700). We appreciate the comments of the reviewers that led to a better presentation.

References

1. Achdou, Y., Nataf, F.: Preconditioners for the mortar method based on local approximations of the Steklov-Poincaré operator. Mathematical Models and Methods in Applied Sciences **5**(7), 967–997 (1995)
2. Arioli, M., Kourounis, D., Loghin, D.: Discrete fractional Sobolev norms for domain decomposition preconditioning. IMA J. Numer. Anal. **33**(1), 318–342 (2013)
3. Arioli, M., Loghin, D.: Discrete interpolation norms with applications. SIAM J. Numer. Anal. **47**(4), 2924–2951 (2009)
4. Bjorstad, P.E., Widlund, O.B.: Iterative methods for the solution of elliptic problems on regions partitioned into substructures. SIAM J. Numer. Anal. **23**(6), 1097–1120 (1986)
5. Bramble, J.H., Pasciak, J.E., Xu, J.: A multilevel preconditioner for domain decomposition boundary systems. In: Proceedings of the 10th international conference on computing methods in applied sciences and engineering on Computing methods in applied sciences and engineering, Paris, pp. 107–118. Nova Science Publishers (1991)
6. Darbas, M., Darrigrand, E., Lafranche, Y.: Combining analytic preconditioner and fast multipole method for the 3-D Helmholtz equation. Journal of Computational Physics **236**, 289 – 316 (2013)
7. Dryja, M.: A capacitance matrix method for Dirichlet problem on polygon region. Numerische Mathematik **39**, 51–64 (1982)
8. Farhat, C., Lesoinne, M., Pierson, K.: A scalable dual-primal domain decomposition method. Numerical Linear Algebra with Applications **7**(7-8), 687–714 (2000)
9. Farhat, C., Mandel, J., Roux, F.X.: Optimal convergence properties of the FETI domain decomposition method. Comput. Methods Appl. Mech. Engrg. **115**, 365–385 (1994)
10. Farhat, C., Roux, F.X.: A method of finite element tearing and interconnecting and its parallel solution algorithm. Int. J. Numer. Meth. Engng. **32**(6), 1205–1227 (1991)
11. Farhat, C., Roux, F.X.: Implicit parallel processing in structural mechanics. Computational Mechanics Advances **II**(1), 1–124 (1994)
12. Hecht, F., Pironneau, O., Morice, J., Hyaric, A., Ohtsuka, K.: Freefem++. Universite Pierre et Marie Curie (2012). URL `http://www.freefem.org/ff++`
13. Krautle, S.: A domain decomposition method using efficient interface-acting preconditioners. Math. Comp. **74**, 1231–1256 (2004)
14. Mandel, J., Tezaur, R.: Convergence of a substructuring method with Lagrange multipliers. Numerische Mathematik **73**(4), 473–487 (1996)
15. Rixen, D.J., Farhat, C.: A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. Int. J. Numer. Meth. Engng **44**(4), 489–516 (1999)
16. Xu, J., Zhang, S.: Preconditioning the Poincaré-Steklov operator by using Green's function. Math. Comput. **66**(217), 125–138 (1997)

Domain Decomposition method for Reaction-Diffusion Systems

Rodrigue Kammogne[1] and Daniel Loghin[1]

1 Introduction

Reaction diffusion systems have important applications in the area of modern mathematical modeling. They can be found in a number of real-life problems, ranging from chemical and biological phenomena to medicine, for example [5, 10]. However the numerical solution to reaction-diffusion problems remains a challenge, as they are often represented as a system of nonlinear PDEs, which are solved on a complex domain. One approach to attempt to solve such problems is to use domain decomposition methods (**DD**), which are more powerful and flexible. They deal with the problem in a more elegant and efficient way, by dividing the domain into subdomains and then obtaining the solution by solving smaller problems on these subdomains.
In a recent paper, Caetano et al. [3] have introduced a non-overlapping domain decomposition algorithm of Schwarz waveform relaxation type for semilinear reaction-diffusion equations. For solving the interface problem they proposed a new type of nonlinear transmission, using Robin or Ventcell transmission conditions, which leads to a solution technique independent of the mesh parameter. However, this has not been extended to reaction-diffusion systems. Our aim in this work is to present an alternative approach to approximate the Steklov-Poincaré operators arising from a non-overlapping **DD**-algorithm for reaction diffusion systems. Our approach is related to that in [2]. The coercivity and the continuity of the Steklov-Poincaré operators arising in a non-overlapping domain decomposition algorithm for scalar elliptic problems with respect to Sobolev norms of index $1/2$ allow us to construct a new interface preconditioner, which leads to solution techniques independent of the mesh size h. We validate the theoretical results on various numerical experiments.

2 Problem Description

Let $\Omega \subset \mathbb{R}^2$ be an open bounded set. We consider the following model problem:

$$\begin{cases} -\mathrm{D}\Delta\mathbf{u} + \mathrm{M}\mathbf{u} = \mathbf{f} & \text{in } \Omega, \\ \mathbf{u} = \mathbf{0} & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where:

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad \mathrm{M} = \begin{pmatrix} \alpha_1(x,y) & \beta_1(x,y) \\ \beta_2(x,y) & \alpha_2(x,y) \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}, \quad \mathrm{D} = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix}.$$

[1] University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom, e-mail: kammognr@for.mat.bham.ac.uk, e-mail: d.loghin@bham.ac.uk

We assume that f_1 and f_2 are in $L^2(\Omega)$ and M satisfies the following bounds for all $(x,y) \in \Omega$:

$$0 < \gamma_{min} < \frac{\xi^T \mathrm{M} \xi}{\xi^T \xi} \quad \text{for all} \quad \xi \in \mathbb{R}^2 \setminus \{0\} \quad \text{and} \quad \|M\| < \gamma_{max}. \tag{2}$$

The weak formulations of problem (1) reads:

$$\begin{cases} \text{Find } \mathbf{u} \in H_0^1(\Omega) \times H_0^1(\Omega) \text{ such that for all } \mathbf{z} \in H_0^1(\Omega) \times H_0^1(\Omega) \\ B(\mathbf{u},\mathbf{z}) = <\mathbf{f},\mathbf{z}>, \end{cases} \tag{3}$$

where:

$$B(\mathbf{w},\mathbf{z}) = \int_\Omega \mathrm{D}\nabla\mathbf{w} : \nabla\mathbf{z} + (\mathrm{M}\mathbf{w}) \cdot \mathbf{z}\, dx, \text{ and } <\mathbf{f},\mathbf{z}> = \int_\Omega \mathbf{f} \cdot \mathbf{z}\, dx.$$

For the weak form (3), it can be shown that the conditions of the Lax-Milgram lemma are satisfied (see [4] for more details). In particular,

$$B(\mathbf{u},\mathbf{z}) \leq \max\{1,\gamma_{max}\}\|\mathbf{u}\|_1\|\mathbf{z}\|_1, \quad \forall \mathbf{u},\mathbf{z} \in H_0^1(\Omega) \times H_0^1(\Omega), \tag{4}$$

$$B(\mathbf{z},\mathbf{z}) \geq \min\{1,\gamma_{min}\}\|\mathbf{z}\|_1^2, \quad \forall \mathbf{z} \in H_0^1(\Omega) \times H_0^1(\Omega) \tag{5}$$

Let $V^h \times V^h$ be a finite dimensional subspace of $H_0^1(\Omega) \times H_0^1(\Omega)$. The finite element discretizations of the weak formulation (3) reads:

$$\begin{cases} \text{Find } \mathbf{u}_h \in V^h \times V^h \text{ such that for all } \mathbf{z}_h \in V^h \times V^h \\ B(\mathbf{u}_h,\mathbf{z}_h) = \langle \mathbf{f}_h, \mathbf{z}_h \rangle. \end{cases} \tag{6}$$

Since (4), (5) hold for all $\mathbf{u}$, $\mathbf{z} \in H_0^1(\Omega) \times H_0^1(\Omega)$, the existence and uniqueness of the solution of formulation (6) is guaranteed by the Lax-Milgram lemma for all $\mathbf{u}_h$, $\mathbf{z}_h \in V^h \times V^h$.

3 Domain decomposition

Let Ω be partitioned into N subdomains without overlap such that:

$$\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \ (i \neq j), \quad \Gamma_i = \partial\Omega_i \backslash \partial\Omega, \quad \Gamma = \bigcup_{i=1}^{N} \Gamma_i.$$

Let also $\mathbf{u}_i = \mathbf{u}\,|_{\Omega_i}$ be the restriction of the solution $\mathbf{u}$ to subdomain Ω_i, and $\mathbf{u}_i\,|_{\Gamma_i} = \boldsymbol{\lambda}_i$ the trace of $\mathbf{u}$ on each interface.
Problem (1) is equivalent to a set of N subproblems:

$$\begin{cases} \mathscr{L}\mathbf{u}_i = \mathbf{f} & \text{in } \Omega_i, \\ \mathbf{u}_i = \mathbf{0} & \text{on } \partial\Omega_i \cap \partial\Omega, \\ \mathbf{u}_i = \boldsymbol{\lambda_i} & \text{on } \Gamma_i, \end{cases} \tag{7}$$

where $\mathscr{L} := -\mathrm{D}\Delta + \mathrm{M}$. If we write $\mathbf{u}_i = \mathbf{w}_i + \mathbf{v}_i$, then equations (7) are equivalent to the following two sets of N subproblems:

$$\begin{cases} \mathscr{L}\mathbf{w}_i = \mathbf{f} \text{ in } \Omega_i; \\ \mathbf{w}_i = \mathbf{0} \text{ on } \partial\Omega_i \cap \partial\Omega; \\ \mathbf{w}_i = \mathbf{0} \text{ on } \Gamma_i; \end{cases} \quad (8) \qquad \begin{cases} \mathscr{L}\mathbf{v}_i = \mathbf{0} \text{ in } \Omega_i; \\ \mathbf{v}_i = \mathbf{0} \text{ on } \partial\Omega_i \cap \partial\Omega; \\ \mathbf{v}_i = \boldsymbol{\lambda_i} \text{ on } \Gamma_i. \end{cases} \quad (9)$$

We can view $\mathbf{v}_i$ as the $\mathscr{L}$-extension of $\boldsymbol{\lambda}_i$ to the domain Ω_i and will be denoted by $H_i\boldsymbol{\lambda}_i$. The equation for $\boldsymbol{\lambda}$ can be shown to be of the form:

$$\sum_{i=1}^{N} \int_{\Gamma_i} (\mathbf{n}_i \cdot \nabla H_i\boldsymbol{\lambda}_i) \cdot \mathbf{z}_i \, ds = -\sum_{i=1}^{N} \int_{\Gamma_i} (\mathbf{n}_i \cdot \nabla \mathbf{w}_i) \cdot \mathbf{z}_i \, ds. \quad (10)$$

From (10), the Steklov-Poincaré operator $\mathscr{S}$ can be defined in the following way:

$$\langle \mathscr{S}\boldsymbol{\lambda}, \boldsymbol{\mu} \rangle := \sum_{i=1}^{N} \int_{\Gamma_i} (\mathbf{n}_i \cdot \nabla H_i\boldsymbol{\lambda}_i) \cdot \boldsymbol{\mu}_i \, ds. \quad (11)$$

The systems (8) and (9) together with the Steklov-Poincaré problem (10) represent the multi-domain formulation of the problem (1).

3.1 Mixed finite element discretisation

The weak formulation of the multi-domain formulation of the problem (1) reads:

$$(1) \begin{cases} \text{Find } \mathbf{u}_i \in H_0^1(\Omega_i) \times H_0^1(\Omega_i) \text{ such that for all } \mathbf{z}_i \in H_0^1(\Omega_i) \times H_0^1(\Omega_i); \\ B_i(\mathbf{w}_i, \mathbf{z}_i) = (\mathbf{f}_i, \mathbf{z}_i). \end{cases}$$

$$(2) \begin{cases} \text{Find } \boldsymbol{\lambda} \in H_{00}^{1/2}(\Gamma) \times H_{00}^{1/2}(\Gamma) \text{ such that for all } \boldsymbol{\eta} \in H_{00}^{1/2}(\Gamma) \times H_{00}^{1/2}(\Gamma); \\ s(\boldsymbol{\lambda}, \boldsymbol{\eta}) := \langle \mathscr{S}\boldsymbol{\lambda}, \boldsymbol{\eta} \rangle = \sum_{i=1}^{N} [(\mathbf{f}_i, \boldsymbol{\eta}_i) - B_i(\mathbf{w}_i, \boldsymbol{\eta}_i)]. \end{cases}$$

$$(3) \begin{cases} \text{Find } \widetilde{\mathbf{v}}_i \in H_0^1(\Omega_i) \times H_0^1(\Omega_i) \text{ such that for all } \mathbf{z}_i \in H_0^1(\Omega_i) \times H_0^1(\Omega_i); \\ B_i(\widetilde{\mathbf{v}}_i, \mathbf{z}_i) = B_i(\mathbf{v}_i, \mathbf{z}_i) - B_i(\mathbf{p}_i, \mathbf{z}_i) = -B_i(\mathbf{p}_i, \mathbf{z}_i). \end{cases}$$

Note that $\widetilde{\mathbf{v}}_i = \mathbf{v}_i - \mathbf{p}_i$, where $\mathbf{p}_i$ is an $\mathscr{L}$-extension of $\boldsymbol{\lambda}_i$ to Ω_i satisfying $p_i = 0$ on $\partial\Omega_i \cap \partial\Omega$.

Let $\mathfrak{T}_h$ denote a subdivision of $\Omega \subset \mathbb{R}^2$ into simplices. We define $V^h = \bigcup_{i=1}^{N} V_i^h$ a subset of $H_0^1(\Omega)$ to be a space of piecewise polynomial functions on $\mathfrak{T}_h$ such that:

$$V_i^h = V_i^{h,r} := \left\{ w \in C^0(\Omega_i) : w|_{\mathfrak{t}} \in P_r \ \ \forall \mathfrak{t} \in \mathfrak{T}_h, \ w\,|_{\partial\Omega \cap \partial\Omega_i} = 0 \right\}.$$

Here $P_r(\mathfrak{t})$ is considered as the space of polynomials in d variables of degree r defined on a set $\mathfrak{t} \subset \mathbb{R}^d$. Given a basis $\{\boldsymbol{\phi}_k\}_{k=1}^{n}$ of $V^h \times V^h$, such that:

$$\mathbf{u}_h(\mathrm{x}) = \sum_{k}^{2(n_I+n_\Gamma)} u_k \boldsymbol{\phi}_k(\mathrm{x}),$$

we obtain the following linear system:

$$\begin{pmatrix} A_{II}^{(1)} & A_{I\Gamma}^{(1)} & M_{II}^{(1)} & M_{I\Gamma}^{(1)} \\ A_{\Gamma I}^{(1)} & A_{\Gamma\Gamma}^{(1)} & M_{\Gamma I}^{(1)} & M_{\Gamma\Gamma}^{(1)} \\ M_{II}^{(2)} & M_{I\Gamma}^{(2)} & A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ M_{\Gamma I}^{(2)} & M_{\Gamma\Gamma}^{(2)} & A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(2)} \end{pmatrix} \begin{pmatrix} u_{1I} \\ u_{1\Gamma} \\ u_{2I} \\ u_{2\Gamma} \end{pmatrix} = \begin{pmatrix} f_{1I} \\ f_{1\Gamma} \\ f_{2I} \\ f_{2\Gamma} \end{pmatrix}; \tag{12}$$

with $A^{(i)} := d_i L + \alpha_i M$ and $M^{(i)} := \beta_i M$. The matrix M is known as the mass matrix, while L represents the discrete Laplacian matrix. We also denote by $S_{A^{(i)}} := A_{\Gamma\Gamma}^{(i)} - A_{\Gamma I}^{(i)}(A_{II}^{(i)})^{-1}A_{I\Gamma}^{(i)}$ the corresponding local Schur complement associated with $A^{(i)}$. Equation (12) can be rewritten as:

$$A\mathbf{u} = \begin{pmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{f}_I \\ \mathbf{f}_\Gamma \end{pmatrix}, \tag{13}$$

where:

$$A_{\mu\nu} = \begin{pmatrix} d_1 L_{\mu\nu} + \alpha_1 M_{\mu\nu} & \beta_1 M_{\mu\nu} \\ \beta_2 M_{\mu\nu} & d_2 L_{\mu\nu} + \alpha_2 M_{\mu\nu} \end{pmatrix}, \quad \mu,\nu \in \{I,\Gamma\}.$$

4 A blockdiagonal interface preconditioner

Let $H_{00}^{1/2}(\Gamma)$ denote the interpolation space between $H_0^1(\Gamma)$ and $L^2(\Gamma)$, which is equipped with the norm $\|\,.\,\|_{1/2,\Gamma}$ as given in [8, chapter 1]. It can be shown that the finite element matrix representation of the norm $\|\,.\,\|_{1/2,\Gamma}$ is given by [1]

$$H_{1/2} := [M_\Gamma, L_\Gamma]_{1/2} := M_\Gamma (M_\Gamma^{-1} L_\Gamma)^{1/2},$$

where M_Γ and L_Γ represent respectively the Mass matrix and discrete Laplacian matrix assembled on Γ. It has been proven in [6] that the matrix $H_{1/2}^{(i)}(\Gamma)$

$$H_{1/2}^{(i)}(\Gamma) := [M_\Gamma, A_\Gamma^{(i)}]_{1/2} := M_\Gamma (M_\Gamma^{-1} A_\Gamma^{(i)})^{1/2}$$

is spectrally equivalent to $H_{1/2}$ for $i = 1,2$, where $A_\Gamma^{(i)} := d_i L_\Gamma + \alpha_i M_\Gamma$.
Consider the following eigenvalue problem:

$$\begin{pmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \end{pmatrix} = \mu \begin{pmatrix} A_{II} & A_{I\Gamma} \\ 0 & P_S \end{pmatrix} \begin{pmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \end{pmatrix} \tag{14}$$

with $S = A_{\Gamma\Gamma} - A_{\Gamma I}A_{II}^{-1}A_{\Gamma I}$. Then $\mu = 1$ or it satisfies:

$$S\mathbf{u}_\Gamma = \mu P_S \mathbf{u}_\Gamma.$$

Using the definition of $\mathscr{S}$ in equation (11), we can derive the following theorem:

Theorem 1. *There exist positive constants c_1, c_2 such that for all $\boldsymbol{\lambda}_h, \boldsymbol{\mu}_h \in H_{00}^{1/2}(\Gamma) \times H_{00}^{1/2}(\Gamma)$:*

$$c_1 \|\boldsymbol{\lambda}_h\|_{1/2,\Gamma}^2 \leq \langle \mathscr{S}\boldsymbol{\lambda}_h, \boldsymbol{\lambda}_h \rangle, \quad \langle \mathscr{S}\boldsymbol{\lambda}_h, \boldsymbol{\mu}_h \rangle \leq c_2 \|\boldsymbol{\lambda}_h\|_{1/2,\Gamma} \|\boldsymbol{\mu}_h\|_{1/2,\Gamma}.$$

Proof. The reader should refer to [6].

From the equivalence between the continuous and the discrete interpolation norms of index $1/2$, we have:

$$\kappa_1 \|\eta_h\|_{1/2,\Gamma} \leq \|\boldsymbol{\eta}\|_{H_{1/2}} \leq \kappa_2 \|\eta_h\|_{1/2,\Gamma}, \quad \forall \boldsymbol{\eta} \in \mathbb{R}^{n_\Gamma}.$$

Therefore we can derive the following inequalities:

Corollary 1. *There exist positive constants $c_1, c_2, \kappa_1, \kappa_2$ such that for all $\boldsymbol{\lambda}, \boldsymbol{\mu} \in \mathbb{R}^{n_\Gamma}$:*

$$\frac{c_1}{\kappa_2^2} \|\boldsymbol{\lambda}\|_{H_{1/2}^{(1)} \oplus H_{1/2}^{(2)}}^2 \leq \langle S\boldsymbol{\lambda}, \boldsymbol{\lambda} \rangle, \quad \langle S\boldsymbol{\lambda}, \boldsymbol{\mu} \rangle \leq \frac{c_2}{\kappa_1^2} \|\boldsymbol{\lambda}\|_{H_{1/2}^{(1)} \oplus H_{1/2}^{(2)}} \|\boldsymbol{\mu}\|_{H_{1/2}^{(1)} \oplus H_{1/2}^{(2)}}.$$

This leads to the following two remarks:

Remark 1. It can be shown using a standard GMRES convergence based on the Field of Values that any symmetric positive definite preconditioner P_S which satisfies:

$$\xi_2 \|\boldsymbol{\lambda}\|_{P_S}^2 \leq \langle S\boldsymbol{\lambda}, \boldsymbol{\lambda} \rangle, \quad \langle S\boldsymbol{\lambda}, \boldsymbol{\mu} \rangle \leq \xi_1 \|\boldsymbol{\lambda}\|_{P_S} \|\boldsymbol{\mu}\|_{P_S}, \ \forall \boldsymbol{\lambda}, \boldsymbol{\mu} \in \mathbb{R}^n,$$

leads to convergence independent of the size of the problem [9].

Remark 2. It has been shown in [6], that there exist constants σ_i, δ_i such that for all $\widetilde{\lambda}, \widetilde{\mu} \in H_{00}^{1/2}(\Gamma)$:

$$\sigma_i \|\widetilde{\lambda}\|_{H_{1/2}^{(i)}}^2 \leq \langle S_{A^{(i)}} \widetilde{\lambda}, \widetilde{\lambda} \rangle, \quad \langle S_{A^{(i)}} \widetilde{\lambda}, \widetilde{\mu} \rangle \leq \delta_i \|\widetilde{\lambda}\|_{H_{1/2}^{(i)}} \|\widetilde{\mu}\|_{H_{1/2}^{(i)}}; \quad i = 1, 2.$$

Then, a natural choice for P_S is:

$$\widehat{S}_1 = \begin{pmatrix} S_{A^{(1)}} & 0 \\ 0 & S_{A^{(2)}} \end{pmatrix}.$$

Another more practical choice for P_S is:

$$\widehat{S}_2 = \begin{pmatrix} H_{1/2}^{(1)} & 0 \\ 0 & H_{1/2}^{(2)} \end{pmatrix}.$$

The implementation of this preconditioner can be achieved using sparse linear algebra techniques. In particular the action of the inverse of $H_{1/2}^{(i)}$ on a given vector $\mathbf{z} \in \mathbb{R}^n$ can be approximated via a generalised Lanczos algorithms (see [1, 2]), which would only involve sparse computations with interface mass and Laplacian matrices.

5 Numerical results

In this section we present the numerical experiments obtained by solving some reaction diffusion problems in two dimensions. All the problems are solved on a square domain $\Omega = (-1,1)^2$. The domain Ω is divided into $N = N_x \times N_y$ subdomains of size $2/N_x \times 2/N_y$ each, with $N_x = N_y \in \{2,4,8\}$. Furthermore, we used a uniform triangulation on each subdomain so that we work with a sequence of nested grids as well as nested subdomain partitions. The GMRES method is employed with a tolerance of 10^{-6} together with the following right preconditioners:

$$P_{R_j} = \begin{pmatrix} A_{II} & A_{I\Gamma} \\ 0 & \widehat{S}_j \end{pmatrix} \quad (j = 1,2).$$

5.1 Test problem 1

We consider now the problem (1), with the following parameters:

$$d_1 = d_2 = 1, \alpha_1 = \alpha_2 = 10^{k_1}, \beta_1 = \beta_2 = 1$$

with $\mathbf{f}$ such that $\mathbf{u}^T = ((x - \frac{1}{3}x^3)(y - \frac{1}{3}y^3), (x - \frac{1}{3}x^3)(y - \frac{1}{3}y^3) + 2)$. We showed in Table 1 that P_{R_1} is an optimal preconditioner for problem (1), as the number of iterations is independent of the problem size and the number of subdomains. However, it remains computationally expensive. A more practical option is P_{R_2}. We find indeed that working with P_{R_2} still gives us virtually no dependence on the size of the problem but a dependence on the number of subdomains. However this dependence disappears for increasing α_i. This latter property is due to the fact that the problem becomes *'easier'* to solve iteratively as the mass matrix becomes more and more dominant. For the remaining test problems, we consider only P_{R_2}.

Preconditioner=	P_{R_1}						P_{R_2}					
k_1=	1			2			3			1			2			3		
domains =	4	16	64	4	16	64	4	16	64	4	16	64	4	16	64	4	16	64
size = 8,450	4	4	4	4	4	4	4	4	4	14	16	19	13	13	14	11	11	11
33,282	4	4	4	4	4	4	4	4	4	14	16	20	13	13	15	11	11	12
132,098	4	4	4	4	4	4	4	4	4	14	16	20	13	14	15	11	11	12

Table 1 GMRES iterations for Problem 1.

5.2 Test problem 2

We solve the same problem as in the previous example but with $d_1 = 1, d_2 = 0.1$ and $k_2 = 0$. Since $d_1 \neq d_2$ two set of results have been obtained (see Table 2). The first set of results is obtained by applying the preconditioner directly to the problem (1). The second set of results is obtained by applying the preconditioner to a scaled version of problem (1), namely:

$$-\Delta \mathbf{v} + \mathrm{M}\mathrm{D}^{-1}\mathbf{v} = \mathbf{f}, \quad \text{where } \mathbf{v} = \mathrm{D}\mathbf{u}. \tag{15}$$

In both cases we have a logarithmic dependence on the number of subdomains and virtually no dependence on the size of the problem. However the number of iterations remains higher than those seen in test problem 1. This is due to the fact that the preconditioned matrices are no longer symmetric.

	Without Scaling			With Scaling		
k_1=	1	2	3	1	2	3
domains =	4 16 64	4 16 64	4 16 64	4 16 64	4 16 64	4 16 64
size = 8,450	20 24 26	17 19 19	16 17 16	13 14 17	12 12 13	9 11 10
33,282	20 24 27	17 20 21	15 19 19	13 14 18	12 13 13	10 12 11
132,098	20 24 28	18 21 22	15 19 19	14 14 18	12 13 13	10 12 12

Table 2 GMRES iterations for Problem 2 .

Remark 3. The similarity between the second part of the results in Table 1 and Table 2 tells us that the performance of our preconditioner will not be affected if $d_1 << d_2$. In that case the scaled version (15) of the problem is used .

5.3 Test problem 3

Finally we consider problem (1) with $d_1 = 1; d_2 = 0.1; \mathbf{f} = (1,1)^T$ and $\mathbf{u} = 0$ on $\partial\Omega$ together with the following jump coefficients:

$$\alpha_1 = \begin{cases} 1 & \text{if } x^2+y^2 < 1/4 \\ 100 & \text{otherwise} \end{cases}; \quad \alpha_2 = \begin{cases} 100 & \text{if } x^2+y^2 < 1/4 \\ 1 & \text{otherwise} \end{cases}$$

$$\beta_1 = \begin{cases} 0.1 & \text{if } x^2+y^2 < 1/4 \\ 1 & \text{otherwise} \end{cases}; \quad \beta_2 = \begin{cases} 1 & \text{if } x^2+y^2 < 1/4 \\ 0.1 & \text{otherwise} \end{cases}$$

An illustration of the final solution $\mathbf{u}$ is provided in Figure 1, while the iteration count is presented in Table 3. We observe a similar convergence behavior: independence of the problem size and logarithmic dependence on the number of subdomains.

6 Conclusion

We presented a general non-overlapping domain decomposition method for solving a system of coupled reaction-diffusion equations (linear case only). We derived the

domains =	4	16	64
size = 8,450	19	24	28
33,282	18	25	28
132,098	18	26	28

Table 3 GMRES iterations for Problem 3.

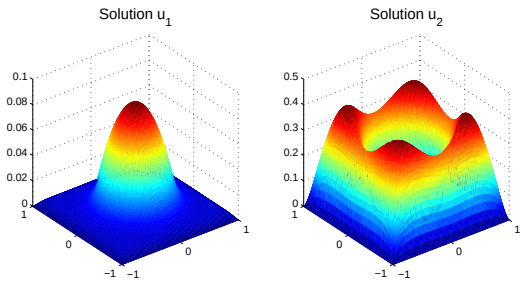


Fig. 1 Solution for Problem 3.

corresponding Steklov-Poincaré operator together with the associated linear algebra problem. In addition, by exploiting the fact that the Steklov-Poincaré operators arising in a non-overlapping **DD**-algorithm are coercive and continuous with respect to Sobolev norms of index $1/2$, an interface preconditioner for the Schur complement problem was constructed, which is strongly related to the finite element representation of the norm $\| \, . \, \|_{1/2,\Gamma}$. Its implementation can be achieved via sparse Lanczos procedures, which do not add to the complexity of the problem . We used various numerical examples to validate our theoretical results. We found that the performance of the method is independent of the mesh size h, but remains at worst logarithmically dependent on the number of subdomains. Similar performance is obtained when using a METIS [7] partitioning of the domain, or when our approach is extended to non-linear reaction-diffusion systems (see [6] for more details).

References

1. M. Arioli and D. Loghin. Discrete interpolation norms with applications. *SIAM Journal on Numerical Analysis*, 47(4):2924–2951, 2009.
2. M. Arioli, D. Kourounis, and D. Loghin. Discrete fractional sobolev norms for domain decomposition preconditioning. *IMA Journal of Numerical Analysis*, 2012.
3. F. Caetano, M.J. Gander, L. Halpern, J. Szeftel, et al. Schwarz waveform relaxation algorithms for semilinear reaction-diffusion equations. *Networks and Heterogeneous Media (NHM)*, 5(3):487–505, 2010.
4. P. G. Ciarlet. *The finite element method for elliptic problems*, volume 4. North Holland, 1978.
5. J. Jiang and J. Shi. Dynamics of a reaction-diffusion system of autocatalytic chemical reaction. *Dynamical systems*, 21(1):245–258, 2008.
6. R. Kammogne and D. Loghin. Domain decomposition for reaction-diffusion systems. *SIAM Journal on Scientific Computing*. In review.
7. G. Karypis and V. Kumar. *METIS-Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0*. 1995.
8. J. L. Lions and E. Magenes. *Non-Homogeneous Boundary Value Problems and Applications, Vol. 1*. Springer-Verlag, Berlin, 1972.
9. D. Loghin and A. J. Wathen. Analysis of preconditioners for saddle-point problems. *SIAM J. Sci. Comput.*, 25(6):2029–2049, June 2004.
10. R. Peng, J. Shi, and M. Wang. On stationary patterns of a reaction–diffusion model with autocatalysis and saturation law. *Nonlinearity*, 21:1471, 2008.
11. A. Toselli and O. Widlund. *Domain decomposition methods-algorithms and theory*, Volume 34. Springer, 2004.

Domain decomposition for the neutron SP_N equations

E. Jamelot[1], P. Ciarlet, Jr.[2], A.-M. Baudron[1], and J.-J. Lautard[1]

1 Introduction

The neutron transport equation allows to describe the neutron flux density in a reactor core. It depends on 7 variables: 3 for the space, 2 for the motion direction, 1 for the energy (or the speed), and 1 for the time. The energy variable is discretized using the multigroup theory [4]. The P_N transport equations are obtained by developing the neutron flux on the spherical harmonics from order 0 to order N. This approach is very time-consuming. The simplified P_N (SP_N) transport theory [14] was developed to address this issue. The two fundamental hypotheses to obtain the SP_N equations are that locally, the angular flux has a planar symmetry; and that the axis system evolves slowly. The neutron flux and the scattering cross sections are then developped on the Legendre polynomials. The order N is odd, and the number of SP_N odd (resp. even) moments is $\frac{N+1}{2}$.
Let $\mathscr{R}$, the domain of studies, be a bounded, open subset of $\mathbb{R}^3$, with a piecewise smooth boundary. Let $G+1$ be the number of energy groups, and let $g \in \{0,..,G\}$. In the time-independent case, the multigroup SP_N equations read in $\mathscr{R}$:

$$\text{Solve in } (\mathbf{p}^g, \phi^g) \mid \begin{cases} \mathbb{T}_o^g \mathbf{p}^g + \mathbf{grad}\,(\,\mathbb{H}\,\phi^g\,) = \sum_{g' \neq g} \mathbb{S}_o^{g'g} \mathbf{p}^{g'}, \\ \mathbb{H}^T \,\mathrm{div}\,\mathbf{p}^g + \mathbb{T}_e^g \,\phi^g = \sum_{g' \neq g} \mathbb{S}_e^{g'g} \phi^{g'} + \frac{1}{\lambda} \chi_g \sum_{g'=0}^{G} \mathbb{M}_f^{g'} \phi^{g'}. \end{cases} \tag{1}$$

For each energy group:
• $\phi^g = (\phi_0^g, \phi_2^g, ...)^T \in \mathbb{R}^{\frac{N+1}{2}}$ (resp. $\mathbf{p}^g = (\mathbf{p}_1^g, \mathbf{p}_3^g, ...)^T \in (\mathbb{R}^3)^{\frac{N+1}{2}}$) denotes the vector containing all the even (resp. odd) moments of the neutron flux.
• $\mathbb{T}_e^g$ (resp. $\mathbb{T}_o^g$) $\in \mathbb{R}^{\frac{N+1}{2} \times \frac{N+1}{2}}$ denotes the even (resp. odd) removal matrix, such that: $\mathbb{T}_e^g = diag\left(\sigma_{r,0}^g, \sigma_{r,2}^g, ...\right)$, $\mathbb{T}_o^g = diag\left(\sigma_{r,1}^g, \sigma_{r,3}^g, ...\right)$, where $\sigma_{r,l}^g$ are proportional to the macroscopic removal cross sections.
• $\mathbb{S}_e^{g'g}$ (resp. $\mathbb{S}_o^{g'g}$) $\in \mathbb{R}^{\frac{N+1}{2} \times \frac{N+1}{2}}$ denotes the even (resp. odd) scattering matrix, such that: $\mathbb{S}_e^{g'g} = diag\left(\sigma_{s,0}^{g' \to g}, \sigma_{s,2}^{g' \to g}, ...\right)$, $\mathbb{S}_o^{g'g} = diag\left(\sigma_{s,1}^{g' \to g}, \sigma_{s,3}^{g' \to g}, ...\right)$, where $\sigma_{s,l}^{g' \to g}$ are proportional to the macroscopic group-transfer cross sections.
• $\mathbb{M}_f^g \in \mathbb{R}^{\frac{N+1}{2} \times \frac{N+1}{2}}$ is such that $(\mathbb{M}_f^g)_{k,l} = \delta_{k,0}\delta_{l,0}\nu_g\sigma_f^g$ (with $\delta_{k,l}$ the Kronecker symbol), so that $\mathbb{M}_f^g \phi^g = (\nu_g \sigma_f^g \phi_0^g, 0, ...)^T$. ν^g is the number of neutrons emitted per fission and σ_f^g the macroscopic fission cross section. χ_g is the fission spectrum.

[1] CEA Saclay, DEN/DANS/DM2S/SERMA/LLPR, F-91191 Gif-sur-Yvette Cedex, e-mail: {Erell.Jamelot}{Anne-Marie.Baudron}{Jean-Jacques.Lautard}@cea.fr .[2] POEMS Laboratory, ENSTA ParisTech, 828, bd des Maréchaux, 91762 Palaiseau Cedex, e-mail: Patrick.Ciarlet@ensta-paristech.fr

• $\mathbb{H} \in \mathbb{R}^{\frac{N+1}{2} \times \frac{N+1}{2}}$ is such that $\mathbb{H}_{k,l} = \delta_{k,l} + \delta_{k,l-1}$.

We must fix boundary conditions (BC) on $\partial\mathscr{R}$, such as Dirichlet BC: $\phi^g = 0$ (zero flux), Neumann BC: $\mathbf{p}^g.\mathbf{n} = 0$ (reflection), or Robin BC (void or isotropic albedo, [2]). From now on, we set zero flux BC.
For simplicity reasons, we will focus on the one-speed SP_N approximation ($G+1=1$). From this study, one can easily deduce the multigroup SP_N case [4], for which we use the Gauss-Seidel method on the energy groups. The group-transfer terms disappear and we can skip the g superscript. We have $\chi_0 = 1$. The linear system (1) corresponds to a set of coupled diffusion equations[1]. Moreover, Eqs (1) can be written in a primal form, with the even moments of the neutron flux as unknowns:

$$-\mathbb{H}^T \mathrm{div}\left(\mathbb{T}_o^{-1}\mathbf{grad}\left(\mathbb{H}\phi\right)\right) + \mathbb{T}_e\,\phi = \frac{1}{\lambda}\mathbb{M}_f\,\phi, \text{ in } \mathscr{R},\ \phi = 0, \text{ on } \partial\mathscr{R}. \tag{2}$$

Due to the structure of Eqs (2), we remark that Eqs (1) actually correspond to a generalized eigenproblem, where λ acts as the inverse of an eigenvalue with associated eigenflux ϕ. One can apply the Krein-Rutman theorem [9] to Eqs (1): the physical solution is necessarily positive, and it is the eigenfunction associated to the largest eigenvalue $k_{eff} = \max_\lambda \lambda$, which is in addition simple. More precisely, k_{eff} characterizes the physical state of the core reactor:
• if $k_{eff} = 1$: The nuclear chain reaction is self-sustaining. The reactor is critical;
• if $k_{eff} > 1$: The chain reaction races. The reactor is supercritical;
• if $k_{eff} < 1$: The chain reaction vanishes. The reactor is subcritical.

2 The one-domain algorithm

As we look for the smallest eigenvalue $(k_{eff})^{-1}$, it can be computed by the inverse power iteration algorithm. After some initial guess is provided, at iteration $m+1$, we deduce $(\mathbf{p}^{m+1}, \phi^{m+1}, k_{eff}^{m+1})$ from $(\mathbf{p}^m, \phi^m, k_{eff}^m)$ by solving Eqs (1) with a source term. Set in a domain $\mathscr{R}$, the inverse power iteration algorithm reads:

Set $(\mathbf{p}^0, \phi^0, k_{eff}^0)$, $m = 0$.
Until convergence, do: $m \leftarrow m+1$

Solve in $(\mathbf{p}^{m+1}, \phi^{m+1})$:

$$\begin{cases} \mathbb{T}_o\,\mathbf{p}^{m+1} + \mathbf{grad}\left(\mathbb{H}\,\phi^{m+1}\right) = 0, \text{ in } \mathscr{R}, \\ \mathbb{H}^T\,\mathrm{div}\,\mathbf{p}^{m+1} + \mathbb{T}_e\,\phi^{m+1} = (k_{eff}^m)^{-1}\mathbb{M}_f\,\phi^m, \text{ in } \mathscr{R}, \\ \phi^{m+1} = 0, \text{ on } \partial\mathscr{R}. \end{cases} \tag{3}$$

Compute: $k_{eff}^{m+1} = k_{eff}^m \int_{\mathscr{R}}(\nu\sigma_f\,\phi_0^{m+1})^2 \,/\, \int_{\mathscr{R}}(\nu\sigma_f\,\phi_0^{m+1}\,\nu\sigma_f\,\phi_0^m)$.

End

[1] Note that the SP_1 equations are similar to the neutron mixed diffusion equations.

Above, the Eqs (3) with unknowns $(\mathbf{p}^{m+1}, \phi^{m+1})$ model the so-called source solver, with a source term equal to $(k_{eff}^m)^{-1} s_f^m$, where $s_f^m = \nu\sigma_f \phi_0^m$. The updated value k_{eff}^{m+1} is inferred as follows: assuming that convergence is achieved, i.e.

$$\mathbb{H}^T \operatorname{div} \mathbf{p}^{m+1} + \mathbb{T}_e \phi^{m+1} = (k_{eff}^{m+1})^{-1} s_f^{m+1}, \tag{4}$$

one can write $(k_{eff}^{m+1})^{-1} s_f^{m+1} = (k_{eff}^m)^{-1} s_f^m$ and, multiplying this equation by s_f^{m+1} and integrating over the domain of computation $\mathscr{R}$, we obtain the equation below (3). The convergence criterion is usually set on $|k_{eff}^{m+1} - k_{eff}^m|$, and $||s_f^{m+1} - s_f^m||$. The inverse power iterations are called the outer iterations as opposed to the inner iterations, which correspond to the iterations of the source solver, with a source S. It reads:

$$\text{Solve in } (\mathbf{p}, \phi): \begin{cases} \mathbb{T}_o \mathbf{p} + \mathbf{grad}\,(\mathbb{H}\phi) = 0, \text{ in } \mathscr{R}, \\ \mathbb{H}^T \operatorname{div} \mathbf{p} + \mathbb{T}_e \phi = S, \text{ in } \mathscr{R}, \\ \phi = 0, \text{ on } \partial\mathscr{R}. \end{cases} \tag{5}$$

In the MINOS solver [1, 2], these equations are solved with Raviart-Thomas-Nédélec FE (RTN FE) on a Cartesian or hexagonal mesh. In order to reduce memory size and time computation, we encoded a DD method to solve (5), studied below.

3 Optimized Schwarz method

In order to use non overlapping subdomains, we chose the Schwarz iterative algorithm with Robin interface conditions to exchange information [11]. Let us split $\mathscr{R}$ in two non-overlapping subdomains $\mathscr{R}_1$ and $\mathscr{R}_2$: $\overline{\mathscr{R}} = \overline{\mathscr{R}}_1 \cup \overline{\mathscr{R}}_2$ such that $\mathscr{R}_1 \cap \mathscr{R}_2 = \emptyset$. We define the interface $\overline{\Gamma} = \overline{\mathscr{R}}_1 \cap \overline{\mathscr{R}}_2$. Let $\mathbf{n}_i$ be the outward unit normal vector to $\partial\mathscr{R}_i$, and $(\mathbf{p}_i, \phi_i) = (\mathbf{p}, \phi)_{|\mathscr{R}_i}$. The Schwarz algorithm reads [5]:

Set $(\mathbf{p}_i^0, \phi_i^0)_{i=1,2}$, $n = 0$.
Until convergence, do: $n \leftarrow n+1$

Solve in $(\mathbf{p}_i^{n+1}, \phi_i^{n+1})_{i=1,2}$:

$$\begin{cases} \mathbb{T}_o \mathbf{p}_i^{n+1} + \mathbf{grad}\left(\mathbb{H}\phi_i^{n+1}\right) = \mathbf{Q}, \text{ in } \mathscr{R}_i,\ i = 1,2, \\ \mathbb{H}^T \operatorname{div} \mathbf{p}_i^{n+1} + \mathbb{T}_e \phi_i^{n+1} = S, \text{ in } \mathscr{R}_i,\ i = 1,2, \\ \phi_i^{n+1} = 0, \text{ on } \partial\mathscr{R}_i \cap \partial\mathscr{R},\ i = 1,2, \\ \mathbf{p}_1^{n+1}.\mathbf{n}_1 + \alpha_1 \phi_1^{n+1} = -\mathbf{p}_2^n.\mathbf{n}_2 + \alpha_1 \phi_2^n, \text{ on } \Gamma, \\ \mathbf{p}_2^{n+1}.\mathbf{n}_2 + \alpha_2 \phi_2^{n+1} = -\mathbf{p}_1^{n(+1)}.\mathbf{n}_1 + \alpha_2 \phi_1^{n(+1)}, \text{ on } \Gamma. \end{cases} \tag{6}$$

End

Here, the Robin parameters are matrices $\alpha_i \in \mathbb{R}^{\frac{N+1}{2} \times \frac{N+1}{2}}$: hence the Robin interface condition can couple all harmonics. The discretization of Eqs (6) with RTN FE is described in [7] for the SP_1 case. Compared to the Schur complement method [10], this method requires less modifications, and rather easy to implement, provided one

has at hand a subdomain solver for the source problem. One has only to ensure the data transfer between the subdomains given by the interface conditions. The $n(+1)$ superscript indicates that we can use either the additive Schwarz method (ASM), or the multiplicative Schwarz method (MSM). We showed in [6, 7] the convergence of the sequences $(\mathbf{p}_i^{n+1}, \phi_i^{n+1})_{i=1,2}$, $n \geq 0$ to $(\mathbf{p}, \phi)_{|\mathscr{R}_i=1,2}$ (in the case $\alpha_1 = \alpha_2$). It is well known that the convergence rate depends highly on the Robin matrices $(\alpha_i)_{i=1,2}$. In order to choose them optimally and automatically, we carried out an asymptotic study, à la Nataf and Nier [12]. For the SP_1 case, we obtained that $\alpha_i = (\sigma_{r,0|\mathscr{R}_j})^{1/2}(\sigma_{r,1|\mathscr{R}_j})^{-1/2}$ [7]. We refer to [6] for the computations of the SP_N case, $N > 1$. In this case, the Robin matrices $(\alpha_i)_{i=1,2}$ are symmetric positive definite, and they depend on the removal cross sections values in $(\mathscr{R}_j)_{j=2,1}$. In the multigroup case, the cross sections depend moreover on the energy groups and so do the $(\alpha_i)_{i=1,2}$. Let us see next how this algorithm modifies the eigenvalue algorithm.

4 The multi-domains algorithm

Applying the Schwarz iterative method to algorithm (3), at iteration $m+1$, we should compute the solution to the source solver iteratively, which yields in principle nested outer ($m \leftarrow m+1$) and inner (index n) iterations. However, numerical experiments show that the inverse power algorithm leads the global convergence: a single inner iteration is sufficient. So, the resulting algorithm contains only one level of iteration (with index m). The inverse power algorithm with DD reads then:

Set $((\mathbf{p}_i^0, \phi_i^0)_{i=1,2}, k_{eff}^0)$, $m = 0$.
Until convergence, do: $m \leftarrow m+1$

Solve in $(\mathbf{p}_i^{m+1}, \phi_i^{m+1})_{i=1,2}$, with $j = 2, 1$:

$$\left\{\begin{array}{rl} \mathbb{T}_o\, \mathbf{p}_i^{m+1} + \mathbf{grad}\left(\mathbb{H}\, \phi_i^{m+1}\right) &= 0, \text{ in } \mathscr{R}_i, \\ \mathbb{H}^T \operatorname{div} \mathbf{p}_i^{m+1} + \mathbb{T}_e\, \phi_i^{m+1} &= (k_{eff}^m)^{-1} \mathbb{M}_f\, \phi_i^m, \text{ in } \mathscr{R}_i, \\ \mathbf{p}_i^{m+1}.\mathbf{n}_i + \alpha_i \phi_i^{m+1} &= -\mathbf{p}_j^{m(+1)}.\mathbf{n}_j + \alpha_i \phi_j^{m(+1)}, \text{ on } \Gamma, \\ \phi_i^{m+1} &= 0, \text{ on } \partial\mathscr{R}_i \cap \partial\mathscr{R}. \end{array}\right. \tag{7}$$

Compute: $k_{eff}^{m+1} = k_{eff}^m \sum_{i=1}^2 \int_{\mathscr{R}_i} (\nu\sigma_f\, \phi_{i,0}^{m+1})^2 \,/\, \sum_{i=1}^2 \int_{\mathscr{R}_i} (\nu\sigma_f\, \phi_{i,0}^{m+1}\, \nu\sigma_f\, \phi_{i,0}^m)$.

End

At iteration $m+1$, convergence is measured on the source, expressed as a vector $\underline{\mathbf{s}}_f$: $\varepsilon_f^{m+1} = \max_{dof} |(\underline{\mathbf{s}}_f^{m+1} - \underline{\mathbf{s}}_f^m)_{dof}| \,/\, (\frac{1}{N} \sum_{dof} |(\underline{\mathbf{s}}_f^{m+1})_{dof}|)$. Iterations stop when $\varepsilon_f^{m+1} \leq \varepsilon_f$, where ε_f is given by the user. Let us test our method.

5 Results

To perform computations, we use the MINOS solver [1, 2] of the APOLLO3®[2] neutronics code. The cross sections come from experimental measurements. They take constant values per unit mesh which can be very different from one mesh to another: we face highly heterogeneous problems. We use the following notations:
• Nc: The number of cores.
• N_{DD}: The $3D$ cartesian (N_{DD}^x, N_{DD}^y ,N_{DD}^z) decomposition.
• N_{out}: The number of outer iterations to achieve convergence.
• $Err.$: The (unsigned) difference between the computed and the converged eigenvalues, either sequentially or in parallel, times 10^{-5}.
• CPU: The CPU time spent within the MINOS solver, given in seconds.
• $Eff.$ (Tab. 3 and 2 only): The efficiency (in %): namely, $T_1/(Nc \times T_N)$, where T_1 is the total sequential CPU time with a single domain, and T_N is the parallel CPU time on Nc cores with Nc subdomains.

For Tab. 1 and 3, we used Intel Xeon L5640 processors with an infiniband network. For Tab. 2, computations were carried out on the Titane computer, hosted by the CCRT (the CEA Supercomputing Center). For each test, we report, above the results Tables, a resulting (x,y) normalized power distribution map of the calculation (Fig. 1, 2, 3).

The results presented in Tab. 1 concern a $3D$ model of a pressurized water reactor (PWR) core of capacity 900 MWe. We performed computations on a mono-core, on the diffusion approximation, with two energy groups ($G+1=2$) and RTN_0 FE. The mesh is of size $(289 \times 289 \times 60)$, which yields more than 40M unknowns. We set $\varepsilon_f = 10^{-5}$. In order to validate our optimization choice, we ran the MSM (with N subdomains), from 1 up to 17 340 subdomains.

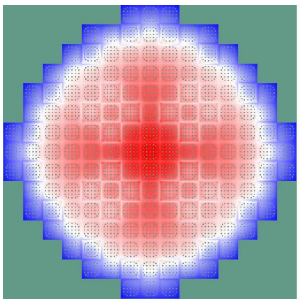


Fig. 1 Power distribution map of the PWR core computation, run with the diffusion approximation, 2 energy groups, RTN_0 FE, MSM.

For $N \leq 4335$, the number of outer iterations does not increase much, and moreover the accuracy is steady. For $N \geq 1156$, the CPU time increase is probably caused by the use of a table to store the subdomains, for which the subdomain access is not

[2] APOLLO3 is a trademark registered in France

Table 1 Results of the PWR core computation (diffusion, 2 energy groups, RTN_0, MSM).

N	N_{DD} (x,y,z)	N_{out}	Err. $\times 10^{-5}$	CPU (s)
1	(1, 1, 1)	381	0.0	230
17	(17, 1, 1)	382	0.0	199
289	(17, 17, 1)	393	0.0	210
1156	(17, 17, 4)	392	0.0	252
2890	(17, 17, 10)	390	0.0	382
4335	(17, 17, 15)	394	0.0	499
8670	(17, 17, 30)	405	0.0	660
17340	(17, 17, 60)	450	0.1	1255

optimized yet. On the other hand, the method seems robust: hence, our optimized choice of the Robin parameters is validated in the diffusion case.

We consider now a $3D$ model of a plate-fuel reactor (PFR) core. We performed computations on the SP_5 approximation, with 4 energy groups ($G+1=4$) and RTN_0 FE. The mesh is of size $364 \times 364 \times 100$, which yields 638M unknowns. We set $\varepsilon_f = 5\,10^{-5}$. We ran the ASM on Nc cores with Nc subdomains.

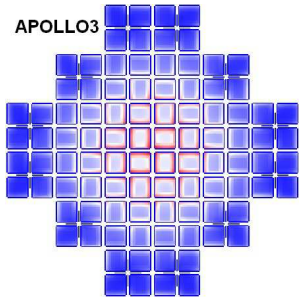


Fig. 2 Power distribution map of the PFR core computation, run with the SP_5 approximation, 4 energy groups, RTN_0 FE, ASM.

Table 2 Results of the PFR core computation (SP_5, 4 energy groups, RTN_0, ASM).

Nc	N_{DD} (x,y,z)	N_{out}	Err. $\times 10^{-5}$	CPU (s)	Eff.
1	(1, 1, 1)	649	0.0	12272	100%
2	(2, 1, 1)	645	0.0	6468	95%
4	(2, 2, 1)	644	0.0	3783	81%
8	(2, 2, 2)	649	0.0	2269	67%
16	(2, 2, 4)	649	0.0	1045	73%
32	(4, 4, 2)	654	0.4	504	76%
64	(4, 4, 4)	643	0.3	303	63%
128	(8, 8, 2)	649	0.2	123	155%

Our DD method converges nicely to the sequential solution, since the error on the eigenvalue is always smaller than $5\,10^{-6}$. Moreover, the number of outer iterations

is quite steady: the optimized choice of the Robin parameters is validated in the SP_N case. The method scales quite well, from 67% up to 155% efficiency on 128 cores. To explain this last result, we suppose that the communication traffic was low, second that some computations were performed in the memory cache.

In [6, 7], we give results which show that choosing random Robin matrices leads to worse results: the number of outer iterations increases faster, and the accuracy deteriorates: in practice, it is important to optimize the Robin matrices.

The last results concern a $2D$ model of the Jules Horowitz reactor (JHR) core[3], dedicated to research, which is currently under construction. We performed computations on the SP_1 approximation, with 6 energy groups ($G+1=6$), and RT_1 FE. The mesh is of size $10^3 \times 10^3$, which represents more than 72M unknowns. We set $\varepsilon_f = 5\,10^{-4}$.

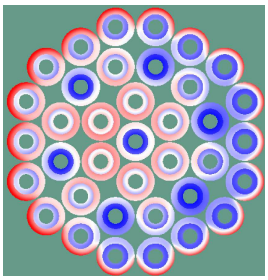


Fig. 3 Power distribution map of the JHR core computation, run with the SP_1 approximation, 6 energy groups, RT_1 FE, ASM.

Table 3 Results of the JHR core computation (SP_1, 6 energy groups, RT_1, ASM).

Nc	N_{DD} (x,y,z)	N_{out}	Err. $\times 10^{-5}$	CPU (s)	Eff.
1	$(1,1)$	639	0.0	1 487	100%
2	$(2,1)$	653	0.4	777	96%
4	$(2,2)$	643	0.5	352	106%
8	$(2,4)$	653	0.1	256	73%
16	$(2,8)$	656	0.2	97	96%
32	$(4,8)$	664	0.6	64	73%
64	$(8,8)$	653	0.9	29	80%

For this last test, the physical geometry is not Cartesian. It probably explains why the accuracy is not as good as for the other tests. The number of outer iterations is quite steady while the efficiency is excellent. In the case of 4 cores, the superlinear efficiency is probably again a consequence of the amount of computations in the memory cache.

[3] `http://www.cad.cea.fr/rjh/index.html`

6 Conclusions and perspectives

We presented a domain decomposition method based on the optimized Schwarz iterative algorithm, to solve the mixed neutrons SP_N equations with RTN FE. Numerical experiments carried out with the MINOS solver show that the method is robust and efficient both sequentially and in parallel, and that our optimized choice of the parameters of the Schwarz algorithm is satisfactory. Note that the number of iterations to solve our problem increases only slightly with the number of subdomains.
Let us finally mention some potential new research directions:
• The use of Ventcell interface conditions: introducing tangential derivatives in the Robin interface condition [12, 8].
• The use of an overlapping DD method with a coarse grid solver, as done in [13].
Finally, let us mention that the MINOS solver can also solve source and kinetic problems [3].

References

1. Baudron, A.-M., Lautard, J.-J.: MINOS: A Simplified P_N solver for core calculations. Nuclear Science and Engineering **155**, 250–263 (2007)
2. Baudron, A.-M., Lautard, J.-J.: SP_N core calculations in the APOLLO3 System. Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011), Latin American Section (LAS) / American Nuclear Society (ANS) (2011)
3. Baudron, A.-M., Lautard, J.-J., Maday, Y., Mula-Hernandez, O.: Parareal for neutronic core calculations. Twenty-first International Conference on Domain Decomposition Methods (2012)
4. Duderstadt, J. J., Hamilton, L. J.: Nuclear reactor analysis. John Wiley & Sons, Inc. (1976)
5. Guérin, P.: Méthodes de décomposition de domaine pour la formulation mixte duale du problème critique de la diffusion des neutrons. Univ. Paris VI (2007)
6. Jamelot, E., Baudron, A.-M., Lautard, J.-J.: Domain decomposition for the SP_N solver MINOS. Transport Theory and Statistical Physics, **41**, 495–512 (2012)
7. Jamelot, E., Ciarlet, P. Jr: Fast non-overlapping Schwarz domain decomposition methods for solving the neutron diffusion equation. J. Comput. Phys. **241**, 445–463 (2013)
8. Japhet, C., Nataf, F., Rogier, F.: The optimized order 2 method: application to convection diffusion problems. Future Generation Computer Systems **18**, 18–30 (2001)
9. Krein, M. G., Rutman, M. A.: Linear operators leaving invariant a cone in a Banach space. Amer. Math. Soc. Translation, Ser. 1, **10**, Functional analysis and measure theory, 199–325 (1962)
10. Lathuilière, B.: Méthodes de décomposition de domaine pour les équations du transport simplifié en neutronique. Univ. Bordeaux I (2010)
11. Lions, P.-L.: On the Schwarz alternating method III: a variant for nonoverlapping subdomains. Third International Symposium Domain Decomposition Methods for Partial Differential Equations (1990)
12. Nataf, F., Nier, F.: Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains. Numer. Math. **75**, 357–377 (1997)
13. Nataf, F. , Xiang, H., Dolean, V.: A two level domain decomposition preconditioner based on local Dirichlet-to-Neumann maps. C. R. Acad. Sci. Paris, Ser. I, **348**, 1163–1167 (2010)
14. Pomraning, G. C.: Asymptotic and variational derivations of the simplified PN Equations. Ann. Nucl. Energy **20**, 9, 623–637 (1993)

A Stochastic-based Optimized Schwarz Method for the Gravimetry Equations on GPU Clusters

Abal-Kassim Cheik Ahamed[1] and Frédéric Magoulès[1]

1 Introduction

By giving another way to see beneath the Earth, gravimetry refines geophysical exploration. In this paper, we evaluate the gravimetry field in the Chicxulub crater area located in between the Yucatan region and the Gulf of Mexico which shows strong gravimetry and magnetic anomalies. High order finite elements analysis is considered with input data arising from real measurements. The linear system is then solved with a domain decomposition method, namely the optimized Schwarz method. The principle of this method is to decompose the computational domain into smaller subdomains and to solve the equations on each subdomain. Each subdomain could easily be allocated to one single processor (i.e. the CPU), each iteration of the optimized Schwarz method involving the solution of the equations on each subdomain (on the GPU). Unfortunately, to obtain high speed-up, several tunings and adaptations of the algorithm should be carrefully performed, such as data transfers between CPU and GPU, and data structures, as described in [3, 2].

The plan of the paper is the following. In Section 2, we present the gravimetry equations. In Section 3, we introduce the optimized Schwarz method, followed in Section 4 by a new idea of using a stochastics-based algorithm to determine the optimized transmission conditions. An overview to the GPU programming model and hardware configuration suite is given in Section 5 for readers not familiar with GPU programming. Section 6 shows numerical experiments which clearly confirm the robustness, competitiveness and efficiency of the proposed method on GPU clusters for solving the gravimetry equations.

2 Gravimetry equations

The gravity force is the resultant of the gravitational force and the centrifugal force. The gravitational potential of a spherical density distribution is equal to $\Phi(r) = Gm/r$, with m the mass of the object, r the distance to the object and G the universal gravity constant equal to $G = 6.672 \times 10^{-11} m^3 kg^{-1} s^{-2}$. The gravitational potential at a given position x initiated by an arbitrary density distribution ρ is given by $\Phi(x) = G \int (\rho(x')/||x - x'||) dx'$ where x' represents one point position within the density distribution. In this paper, we consider only regional scale of the

[1] Ecole Centrale Paris, France, e-mail: akcheik@gmail.com, frederic.magoules@hotmail.com

gravimetry equations therefore we do not take into account the effects related to the centrifugal force. The gravitational potential Φ of a density anomaly distribution $\delta\rho$ is thus given as the solution of the Poisson equation $\Delta\Phi = -4\pi G\delta\rho$.

3 Optimised Schwarz method

The classical Schwarz algorithm was invented more than a century ago [16] to prove existence and uniqueness of solutions to Laplace's equation on irregular domains. Schwarz decomposed the irregular domain into overlapping regular ones and formulated an iteration which used only solutions on regular domains and which converged to a unique solution on the irregular domain. The convergence speed of the classical Schwarz algorithm is proportional to the size of the overlap between the subdomains. A variant of this algorithm can be formulated with non-overlapping subdomains and the transmission conditions should be changed from Dirichlet to Robin [6]. These absorbing boundary transmission conditions defined on the interface between the non-overlapping subdomains, are the key ingredients to obtain a fast convergence of the iterative Schwarz algorithm [5, 9]. Optimal transmission conditions can be derived but consists of non local operators and thus are not easy to implement in a parallel computational environment. One alternative is to approximate these operators with partial differential operators. This paper investigates an approximation based on a new stochastics optimization procedure.

For the sake of clarity, the gravimetry equations are considered in the domain Ω with homogeneous Dirichlet condition. The domain is decomposed into two non-overlapping subdomains $\Omega^{(1)}$ and $\Omega^{(2)}$ with an interface Γ. The Schwarz algorithm can be written as:

$$\begin{aligned}
-\Delta\Phi_{n+1}^{(1)} &= f, \quad \text{in} \quad \Omega^{(1)} \\
\left(\partial_\nu\Phi_{n+1}^{(1)} + \mathscr{A}^{(1)}\Phi_{n+1}^{(1)}\right) &= \left(\partial_\nu\Phi_{n}^{(2)} + \mathscr{A}^{(1)}\Phi_{n}^{(2)}\right), \quad \text{on} \quad \Gamma \\
-\Delta\Phi_{n+1}^{(2)} &= f, \quad \text{in} \quad \Omega^{(2)} \\
\left(\partial_\nu\Phi_{n+1}^{(2)} - \mathscr{A}^{(2)}\Phi_{n+1}^{(2)}\right) &= \left(\partial_\nu\Phi_{n}^{(1)} - \mathscr{A}^{(2)}\Phi_{n}^{(1)}\right), \quad \text{on} \quad \Gamma
\end{aligned}$$

with n the iteration number, and ν the unit normal vector defined on Γ. The operators $\mathscr{A}^{(1)}$ and $\mathscr{A}^{(2)}$ are to be determined for best performance of the algorithm. Considering the case $\Omega = \mathbb{R}^2$, $f = 0$, and applying a Fourier transform, similar calculations as in [7] lead to the expression of the Fourier convergence rate, involving the quantities $\Lambda^{(1)}$ and $\Lambda^{(2)}$, which are the Fourier transforms of $\mathscr{A}^{(1)}$ and $\mathscr{A}^{(2)}$ operators. In this case, the choice $\Lambda^{(1)} := |k|$, and $\Lambda^{(2)} := |k|$ is optimal, since with this choice the algorithm converges in two iterations for two subdomains. Different techniques to approximate these non local operators with partial differential operators have been analyzed in recent years [5, 4, 7]. These techniques consist to define partial differential operators involving a tangential derivative on the inter-

face such as: $\mathscr{A}^{(s)} := p^{(s)} + q^{(s)}\partial^2_{\tau^2}$, with s the subdomain number, $p^{(s)}$, $q^{(s)}$ two coefficients, and τ the unit tangential vector. The first results presented in [5, 9] use a zero order Taylor expansion of the non local operators to find $p^{(s)}$ and $q^{(s)}$. In [8] for convection diffusion equations, in [4] for Maxwell equation, in [7, 12] for the Helmholtz equation, and in [11, 10] for heterogeneous media, a minimization procedure has been used. The function to minimize, i.e., the cost function, is the maximum of the Fourier convergence rate for the frequency ranges considered, and the approach consists to determine the free parameters $p^{(s)}$ and $q^{(s)}$ through an optimization problem. Despite, analytic expressions of $p^{(s)}$ and $q^{(s)}$ can be determined for some specific problems, finding quasi-optimal coefficients numerically is also a good alternative [13]. Furthemore, since the evaluation of the cost function is quite fast and the dimension of the search space reasonable, a more robust minimization procedure could be considered, in the next section. Extension to non-regular geometry can be performed as described in reference [14].

4 Stochastic-based optimised transmission conditions

The stochastic minimization technique we propose to use now, explores the whole space of solutions and finds absolute minima; this technique is based on the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). This algorithm is very robust [1], has good global search ability and does not need to compute the derivatives of the cost function. This algorithm only needs an initial search zone and a population size, even if the solution can be found outside of the initial search zone. The population size parameter is a trade-off between speed and global search. Meaning that, smaller populations lead to faster execution of the algorithm but have more chance to find a local minimum, and, larger sizes allow to avoid local minima better but need more cost function evaluations. For our purpose, a population size of 25 has been large enough to find the global minimum in a few second or less.

The main idea of the algorithm is to find the minimum of the cost function by iteratively refining a search distribution. The distribution is described as a general multivariate normal distribution $\mathrm{d}(m,C)$. Initially, the distribution is given by the user. Then, at each iteration, λ samples are randomly chosen in this distribution and the evaluation of the cost function at those points is used to compute a new distribution. When the variance of the distribution is small enough, the center of the distribution m is taken as solution. After evaluating the cost function for a new population, the samples are sorted by cost and only the μ best are kept. The new distribution center is computed with a weighted mean (usually, more weight is put on the best samples). The step size σ is a factor used to scale the standard deviation of the distribution, i.e., the variance of the search distribution is proportional to the square of the step size. The step size determines the "size" of the distribution. The covariance matrix C determines the "shape" of the distribution, i.e., it determines the principal directions of the distribution and their relative scaling. Adapting (or updating) the covariance matrix is the most complex part of the algorithm. While

this could be done using only the current population, it would be unreliable especially with a small population size; thus the population of the previous iteration should also been taken into account.

5 Overview of GPU programming model

Parallel computation was generally carried out on Central Processing Unit (CPU) cluster until the apparition in the early 2000s of the Graphics Processing Unit (GPU) that facing the migration of the era of GPU computing. The peak performance of CPUs and GPUs is significanlty different, due to the inherently different architectures between these processors. The first idea behind the architecture of GPU is to have many small floating points processors exploiting large amount of data in parallel. This is achieved through a memory hierarchy that allows each processor to optimally access the requiered data. The gains of GPU computing is significantly higher for large size problem compared to CPU, due to the difference between these two architectures. GPU computing requires using graphics programming languages such as NVIDIA CUDA, or OpenCL. *Compute Unified Device Architecture* (CUDA) [15] programming model is an extension of the C language and has been used in this paper.

A specific characteristic of GPU compared to CPU is the feature of memory used. Indeed, a CPU is constantly accessing the RAM, therefore it has a low latency at the detriment of its raw throughput. CUDA devices have four main types of memory: (i) *Global memory* is the memory that ensures the interaction with the host (CPU), and is not only large in size and off-chip, but also available to all threads (also known as compute units), and is the slowest; (ii) *Constant memory* is read only from the device, is generally cached for fast access, and provides interaction with the host; (iii) *Shared memory* is much faster than global memory and is accessible by any thread of the block from which it was created; (iv) *Local memory* is specific to each compute unit and cannot be used to communicate between compute units.

Threads are grouped into blocks and executed in parallel simultaneously, see Figure 1. A GPU is associated with a *grid*, i.e., all running or waiting blocks in the running queue and a kernel that will run on many cores. An ALU is associated with the thread which is currently processing. Threading is not an automated procedure. The developer chooses for each kernel the distribution of the threads, which are organized (*gridification* process) as follows: (i) threads are grouped into blocks; (ii) each block has three dimensions to classify threads; (iii) blocks are grouped together in a grid of two dimensions. The threads are then distributed to these levels and become easily identifiable by their positions in the grid according to the block they belongs to and their spatial dimensions. The kernel function must be called also providing at least two special parameters: the dimension of the block, *nBlocks*, and the number of threads per block, *nThreadsPerBlock*. Figure 1 presents the CUDA processing flow. Data are first copied from the main memory to the GPU memory, (1). Then the host (CPU) instructs the device (GPU) to carry out calculations, (2).

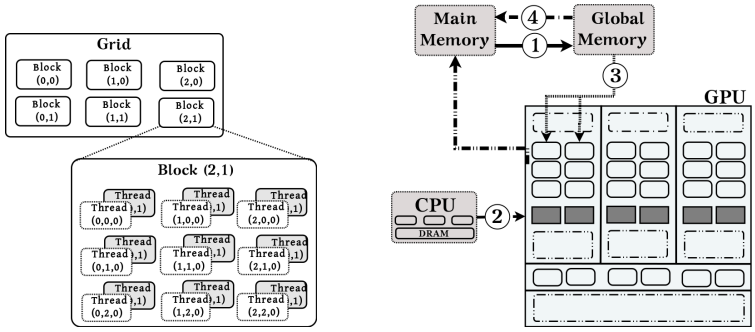


Fig. 1 Gridification of a GPU. Tthread, block, grid (left); GPU computing processing (right)

The kernel is then executed by all threads in parallel on the device, (3). Finally, the device results are copied back (from GPU memory) to the host (main memory), (4). To cope with this difficulty the implementation proposed in this paper uses some advanced tuning techniques developped by the authors, but the details are outside the scope of this paper, and the reader is refeered to [3, 2] for the computer science aspects of this tuning.

6 Numerical analysis

In this section, we report the experiments performed to evaluate the speed-up of our implementation. The Chicxulub impact crater, formed about 65 million years ago, is now widely accepted as the main footprint of the global mass extinction event that marked the Cretaceous/Paleogene boundary. Because of its relevance, in the last two decades, this impact structure has been used as a natural laboratory to investigate impact cratering formation processes and to indirectly infer global effects of large-scale impacts. The crater is buried under 1 km of carbonate sediments in the Yucatan platform. The crater is about 200 km in rim diameter, half on-land and half off-shore with geometric center at Chicxulub Puerto. The internal structure of the Chicxulub crater has been imaged by using several geophysical data sets from land, marine and aerial measurements.

In this paper we perform a finite element analysis of the gravimetry equation using the characteristics of the region provided by the measure. The domain consists of an area of $250 \times 250 \times 15$ in each spatial direction, and is discretized with high order finite element with a total of 19 933 056 degrees of freedom. This mesh is partionned in the x-direction. Each subdomain is allocated to one single processor (i.e. the CPU), each iteration of the optimized Schwarz method involving the solution of the equations inside each subdomain is allocated to one single accelerator (i.e the GPU). We compare the computational time of the optimised Schwarz method using one subdomain per CPU with the optimised Schwarz method using one subdomain

per CPU and a GPU accelerator. For this particular model we performed calculations using our CUDA implementation of the Schwarz method with stochastics-based optimization procedure. The workstation used for all the experiments consists of 1 596 servers Bull Novascale R422Intel Nehalem-based nodes. Each node is composed of 2 processors Intel Xeon 5570 quad-cores (2.93 GHz) and 24 GB of memory (3Go per cores). 96 CPU servers are interconnected with 48 compute Tesla S1070 servers NVIDIA (4 Tesla cards with 4GB of memory by server) and 960 processing units are available for each server.

For the subdomain problems, the diagonal preconditioner conjugate gradient (PCG) is used and the coefficient matrices are stored in CSR format. We fix a residual tolerance threshold of $\varepsilon = 10^{-10}$ for PCG. *Alinea* [3, 2], our research group library, implemented in *C++*, which offers CPU and GPU solvers, is used for solving linear algebra system. In this paper, the GPU is used to accelerate the solution of PCG algorithm. PCG algorithm required the computation of addition of vectors (*Daxpy*), dot product and sparse matrix-vector multiplication. In GPU-implementation considered (Alinea library), the distribution of threads (*gridication*, differs with these operations. The gridification of *Daxpy*, dot product and sparse matrix-vector product correspond respectively to ($nBlocks = \frac{numb\_rows+numb\_th\_block-1}{numb\_th\_block}$, $nThreadsPerBlock = 256$), ($nBlocks = \frac{numb\_rows+numb\_th\_block-1}{numb\_th\_block}$, $nThreadsPerBlock = 128$) and ($nBlocks = \frac{(numb\_rows \times n\_th\_warp)+numb\_th\_block-1}{numb\_th\_block}$, $nThreadsPerBlock = 256$), where *numb_rows*, n_th_warp and $numb\_th\_block$ represent respectively the number of rows of the matrix, the number of threads per warp and the thread block size. We fix for all operations 8 threads per warp. GPU is used only for solving subdomain problems in each iteration. GPU experiment workstation Tesla S1070 has 4 GPUs of 240 cores. The number of computing units depends both on the size of the subdomain problem and the gridification that use 256 threads per threads and 8 threads per warp as introduced in [3, 2].

In our experiments, the CMA-ES algorithm considers as the cost function the Fourier convergence rate of the optimised Schwarz method. We consider for the CMA-ES the following stopping criteria of : a maximum of number iterations equal to 7200 and a residu threshold equal to 5×10^{-11}. Fig. 2 represents the convergence rate of the Schwarz algorithm in the Fourier space, respectively for the symmetric zeroth order (top-left), unsymmetric zeroth order (bottom-left), the symmetric second order (top-right) and unsymmetric second order (bottom-right) transmission conditions obtained from the CMA-ES algorithm. The Fourier convergence rate of the Schwarz method with one side (respectilely two sides) transmission conditions obtained from the CMA-ES algorithm is presented in Fig. 2 and Table 1. The distribution of processors is computed as follows: number of processors $=$ $2 \times$ number of nodes, where 2 corresponds to the number of GPU per node as available on our workstation. As a consequence, only two processors will share the bandwidth, which strongly improve the communications, especially the inter-subdomain communications. Table 2 presents the results done with double precision with a residu threshold, *i.e.* stopping criterion equal to 10^{-6}, for several number of subdomains (one subdomain per processor).

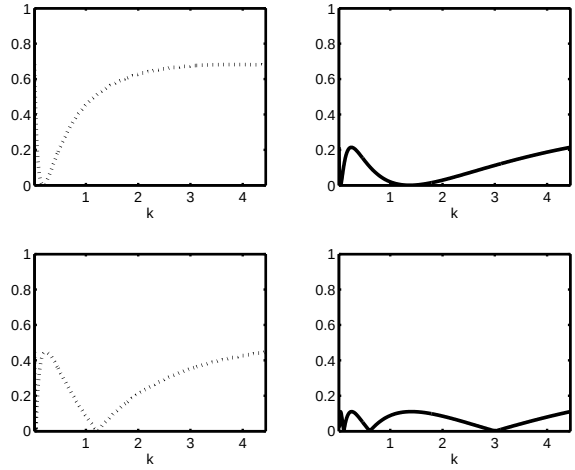


Fig. 2 Fourier convergence rate of the Schwarz algorithm

	$p^{(1)}$	$q^{(1)}$	$p^{(2)}$	$q^{(2)}$	ρ_{max}
oo0_symmetric	0.1826	0	0.1826	0	0.6823
oo0_unsymmetric	1.2193	0	0.0469	0	0.4464
oo2_symmetric	0.0471	0.7050	0.0471	0.7050	0.2143
oo2_unsymmetric	0.1081	0.3205	0.0231	1.5786	0.1101

Table 1 Optimized coefficients obtained from *CMA-ES* algorithm

#subdomains	**#iterations**	**cpu time (sec)**	**gpu time (sec)**	**SpeedUp**
32	41	11 240	1 600	**7.03**
64	45	5 360	860	**6.23**
128	92	6 535	960	**6.81**

Table 2 Comparison of the implementation of our method on CPU and GPU

7 Conclusion

In this paper, we have presented a stochastic-based optimized Schwarz method for the gravimetry equation on GPU Clusters. The effectiveness and robustness of our method are evaluated by numerical experiments performed on a cluster composed of 1 596 servers Bull Novascale R422Intel Nehalem-based nodes where 96 CPU servers are interconnected with 48 compute Tesla S1070 servers NVIDIA (4 Tesla cards with 4GB of memory by server). The presented results range from 32 up-to 128 subdomains show the interest of the use of GPU technologies for solving large size problems, and outline the robustness, performance and efficiency of our Schwarz domain decomposition method with stochastic-based optimized conditions.

Acknowledgements The authors acknowledge partial financial support from the OpenGPU project (2010-2012), and GENCI (Grand Equipement National de Calcul Intensif) for the computer time used during this long-term trend.

References

1. Auger, A., Hansen, N.: Tutorial CMA-ES: evolution strategies and covariance matrix adaptation. In: GECCO (Companion), pp. 827–848 (2012)
2. Cheik Ahamed, A.K., Magoulès, F.: Fast sparse matrix-vector multiplication on graphics processing unit for finite element analysis. In: HPCC-ICESS, pp. 1307–1314. IEEE Computer Society (2012)
3. Cheik Ahamed, A.K., Magoulès, F.: Iterative methods for sparse linear systems on graphics processing unit. In: HPCC-ICESS, pp. 836–842. IEEE Computer Society (2012)
4. Chevalier, P., Nataf, F.: Symmetrized method with optimized second-order conditions for the Helmholtz equation. In: Domain decomposition methods, 10 (Boulder, CO, 1997), pp. 400–407. Amer. Math. Soc., Providence, RI (1998)
5. Després, B.: Domain decomposition method and the Helmholtz problem.II. In: Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE, 1993), pp. 197–206. SIAM, Philadelphia, PA (1993)
6. Després, B., Joly, P., Roberts, J.E.: A domain decomposition method for harmonic Maxwell equations. In: Iterative methods in linear algebra, pp. 475–484. North-Holland, Amsterdam (1992)
7. Gander, M.J., Magoulès, F., Nataf, F.: Optimized Schwarz methods without overlap for the Helmholtz equation. SIAM J. Sci. Comput. **24**(1), 38–60 (2002)
8. Japhet, C., Nataf, F., Rogier, F.: The optimized order 2 method. Application to convection-diffusion problems. Future Generation Computer Systems **18**(1), 17–30 (2001)
9. J.D.Benamou, Després, B.: A domain decomposition method for the Helmholtz equation and related optimal control problems. J. of Comp. Physics **136**, 68–82 (1997)
10. Maday, Y., Magoulès, F.: Improved ad hoc interface conditions for Schwarz solution procedure tuned to highly heterogeneous media. Applied Mathematical Modelling **30**(8), 731–743 (2006)
11. Maday, Y., Magoulès, F.: Optimized Schwarz methods without overlap for highly heterogeneous media. Comput. Methods in Appl. Mech. Eng. **196**(8), 1541–1553 (2007)
12. Magoulès, F., Ivànyi, P., Topping, B.: Convergence analysis of Schwarz methods without overlap for the helmholtz equation. Computers & Structures **82**(22), 1835–1847 (2004)
13. Magoulès, F., Ivànyi, P., Topping, B.: Non-overlapping Schwarz methods with optimized transmission conditions for the Helmholtz equation. Computer Methods in Applied Mechanics and Engineering **193**(45-47), 4797–4818 (2004)
14. Magoulès, F., Putanowicz, R.: Optimal convergence of non-overlapping Schwarz methods for the Helmholtz equation. Journal of Computational Acoustics **13**(3), 525–545 (2005)
15. Nvidia Corporation: CUDA Toolkit Reference Manual, 4.0 edn. Available on line at: `http://developer.nvidia.com/cuda-toolkit-40` (accessed on September 29, 2012)
16. Schwarz, H.A.: ber einen grenzbergang durch alternierendes verfahren. Vierteljahrsschrift der Naturforschenden Gesellschaft **15**, 272–286 (1870)

A parallel preconditioner for a FETI-DP method for the Crouzeix-Raviart finite element

Leszek Marcinkowski*[1] Talal Rahman[2]

1 Introduction

In this paper, we present a Neumann-Dirichlet type parallel preconditioner for a FETI-DP method for the nonconforming Crouzeix-Raviart (CR) finite element discretization of a model second order elliptic problem. The proposed method is almost optimal, in fact, the condition number of the preconditioned problem grows polylogarithmically with respect to the mesh parameters of the local triangulations.

In many scientific applications, where partial differential equations are used to model, the Crouzeix-Raviart (CR) finite element has been one of the most commonly used nonconforming finite element for the numerical solution. This includes applications like the Poisson equation (cf. [11, 23]), the Darcy-Stokes problem (cf. [8]), the elasticity problem (cf. [3]). We also would like to add that there is a close relationship between mixed finite elements and the nonconforming finite element for the second order elliptic problem; cf. [1, 2]. The CR element has also been used in the framework of finite volume element method; cf. [9].

There exists quite a number of works focusing on iterative methods for the CR finite element for second order problems; cf. [4, 5, 10, 13, 16, 18, 19, 20, 21, 22] and references therein. The purpose of this paper is to propose a parallel algorithm based on a Neumann-Dirichlet preconditioner for a FETI-DP formulation of the CR finite element method for the second order elliptic problem. To our knowledge, this is apparently the first work on such preconditioner for the FETI-DP method for the Crouzeix-Raviart (CR) finite element.

The FETI-DP method, which was first introduced in [12], describes a class of fast and efficient domain decomposition solvers for systems of algebraic equations arising from the finite element discretization of elliptic partial differential equations, cf. [17, 14, 15, 24] and references therein.

In a FETI-DP method one has to solve a linear system for a set of dual variables, formulated after eliminating the primal variables. The FETI-DP system contains in itself a coarse problem which is associated with the primal variables, while its preconditioner is based on solving only local problems which is fully parallel.

In this paper, we first present the Crouzeix Raviart discretization of the differential problem, a FETI-DP formulation of the problem is then introduced, and finally a Neumann-Dirichlet preconditioner for the FETI-DP problem is proposed.

Faculty of Mathematics, University of Warsaw, Banacha 2, 02-097 Warszawa, Poland, Leszek.Marcinkowski@mimuw.edu.pl · Department of Computer Engineering, Bergen University College, Nygårdsgaten 112, N-5020 Bergen, Norway, Talal.Rahman@hib.no

* This work was partially supported by Polish Scientific Grant 2011/01/B/ST1/01179.

We present an almost optimal bound for the condition number, showing that the condition number of the preconditioned system grows like $C(1+\log(H/h))^2$, where H is the maximal diameter of the subdomains and h is the fine mesh size parameter.

2 Discrete problem

In this section we present the Crouzeix-Raviart finite element discretization of a model second order elliptic problem with discontinuous coefficients.

Let Ω be a polygonal domain in the plane. We assume that there exists a partition of Ω into disjoint polygonal subdomains Ω_k such that $\overline{\Omega} = \bigcup_{k=1}^N \overline{\Omega}_k$ with $\overline{\Omega}_k \cap \overline{\Omega}_l$ being an empty set, an edge or a vertex (crosspoint). We also assume that these subdomains form a coarse triangulation of the domain which is shape regular in the sense of [7]. We introduce a global interface $\Gamma = \bigcup_i \overline{\partial\Omega_i \setminus \partial\Omega}$ which plays an important role in our study.

Our model differential problem is to find $u^* \in H_0^1(\Omega)$ such that

$$a(u^*, v) = \int_\Omega f v\, dx \quad \forall v \in H_0^1(\Omega), \tag{1}$$

where $f \in L^2(\Omega)$, and $a(u,v) = \sum_{k=1}^N \int_{\Omega_k} \rho_k \nabla u \nabla v\, dx$. The coefficients ρ_k are positive and constant.

We assume that there exists a quasiuniform triangulation, $T_h = T_h(\Omega) = \{\tau\}$, of Ω such that any element τ of T_h is contained in only one subdomain, as a consequence any subdomain Ω_k inherits a local triangulation $T_h(\Omega_k) = \{\tau\}_{\tau \subset \Omega_k, \tau \in T_h}$.

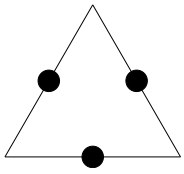


Fig. 1 Illustrating the CR finite element in 2D with black dots as the CR nodal points or CR nodes.

Let $h = \max_{\tau \in T_h(\Omega)} \mathrm{diam}(\tau)$ be the mesh size parameter of the triangulation, cf. [6]. We introduce the following sets of Crouzeix-Raviart (CR) nodal points or -nodes: $\Omega_h^{CR}, \partial\Omega_h^{CR}, \Omega_{k,h}^{CR}, \partial\Omega_{k,h}^{CR}$, and $\Gamma_{kl,h}^{CR}$ correspond to $\Omega, \partial\Omega, \Omega_k, \partial\Omega_k$, and Γ_{kl}, respectively. Here Γ_{kl} is an interface, an open edge, which is shared by the two subdomains, Ω_k and Ω_l.

We now introduce the local finite element spaces. Let $\widehat{W}^h(\Omega)$ be the Crouzeix-Raviart finite element space defined as follows,

$$\widehat{W}^h(\Omega) = \{u \in L^2(\Omega) : u|_\tau \in P_1(\tau) \text{ for each triangle } \tau \in T_h(\Omega), \\ u \text{ is continuous at every midpoint } m \in \Omega_h^{CR} \tag{2}$$

$$\text{and } u(m) = 0 \text{ for every } m \in \partial\Omega_h^{CR}\}.$$

Here $P_1(\tau)$ is the function space of linear polynomials defined over τ. The degrees of freedom of a function $u \in \widehat{W}^h(\Omega)$ over $\tau \in T_h(\Omega)$ are: $\{u(m_j)\}_{j=1,2,3}$, where m_j is a midpoint of an edge of τ, cf. Fig. 1.

We define the local CR space $W^h(\Omega_k)$ as the space of functions which are restrictions to Ω_k of the functions of $\widehat{W}^h(\Omega)$, i.e. $W^h(\Omega_k) = \{u_{|\Omega_k} : u \in \widehat{W}^h(\Omega)\}$. The standard nodal basis function, ϕ_x^{CR}, of $W^h(\Omega_k)$, associated with the CR nodal point $x \in \overline{\Omega}_k^{CR}$, is a function which is equal to one at x and zero at the remaining CR nodal points of $\overline{\Omega}_k^{CR} \setminus \partial\Omega^{CR}$. $\{\phi_x^{CR}\}_{x \in \overline{\Omega}_k^{CR} \setminus \partial\Omega^{CR}}$ is the standard nodal basis of $W^h(\Omega_k)$.

The discrete problem is then defined as follows: Find $u_h^* \in \widehat{W}^h(\Omega)$ such that

$$a_h(u_h^*, v) = f(v) \quad \forall v \in \widehat{W}^h(\Omega), \tag{3}$$

where $a_h(u,v) := \sum_{k=1}^N a_{k,h}(u,v)$ with the local broken bilinear form:

$$a_{k,h}(u,v) := \sum_{\tau \in T_h(\Omega_k)} \int_\tau \rho_k \nabla u \nabla v \, dx.$$

This problem has a unique solution, and an optimal error bound is known; cf. [6].

We shall now reformulate (3) as a saddle point problem. We start by introducing the following global space defined over Ω as follows,

$$W^h(\Omega) := \Pi_{k=1}^N W^h(\Omega_k).$$

Note that each interface Γ_{kl} inherits a 1D triangulation $T_h(\Gamma_{kl})$ from T_h. We define $V^h(\Gamma_{kl})$ as the space of piecewise constant functions over $T_h(\Gamma_{kl})$. In FETI-DP, an important role is played by the global interface which is defined as $\overline{\Gamma} := \bigcup_{k=1}^N \partial\Omega_k \setminus \partial\Omega$. Then, let

$$V^h(\Gamma) := \prod_{\Gamma_{kl} \subset \Gamma} V^h(\Gamma_{kl})$$

be the auxiliary interface space which will be later used as the space of Lagrange multipliers. We introduce the bilinear form $b(u,\psi) : W^h(\Omega) \times V^h(\Gamma) \to \mathbb{R}$ as follows: let $u = (u_k)_{k=1}^N \in W^h(\Omega)$ and $\psi = (\psi_{lk})_{\Gamma_{kl}} \in V^h(\Gamma)$, then $b(u,\psi) = \sum_{\Gamma_{kl} \subset \Gamma} b_{lk}(u, \psi_{lk})$ with

$$b_{lk}(u, \psi_{lk}) = \int_{\Gamma_{kl}} (u_k - u_l)\psi_{lk} \, ds \qquad k > l.$$

Throughout the rest of this paper, we will use the same notation to denote a function and its vector representation with values of the degrees of freedom (dofs) of this function as entries in the representation.

Let c_r be a crosspoint, which is a subdomain vertex, not lying on $\partial\Omega$, and let $\mathscr{V}^{CR}(c_r)$ be the set of CR nodal points of those triangle edges that lie on sub-

domain boundaries and are incident to c_R, e.g. the black dots in Figure 2. Let $\mathscr{V}^{CR} = \bigcup_{c_r \in \Gamma} \mathscr{V}^{CR}(c_r)$.

We then introduce $\widetilde{W}^h(\Omega)$ as the subspace of $W^h(\Omega)$ of functions which are continuous at the CR nodes of $\mathscr{V}^{CR}$. We also introduce a reduced Lagrange multiplier space as follows,

$$\widetilde{V}^h(\Gamma) := \{\lambda \in V^h(\Gamma) : \lambda(m) = 0 \quad \forall m \in \Gamma_h^{CR} \cap \mathscr{V}^{CR}\} \subset V^h(\Gamma).$$

The discrete problem can now be reformulated as the following saddle point prob-

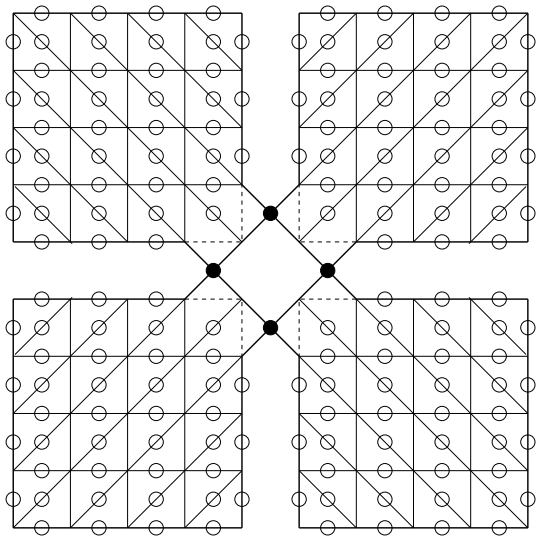


Fig. 2 Illustrating a four subdomain case with one crosspoint. Black dots in the figure represent the CR nodes of $\mathscr{V}^{CR}$ corresponding to the cross point. CR nodes (both circles and black dots) which the degrees of freedom (dofs) of $\widetilde{W}^h(\Omega)$ are associated with, are also shown.

lem: find the pair $(u_h^*, \lambda^*) \in \widetilde{W}^h(\Omega) \times \widetilde{V}^h(\Gamma)$ such that

$$\begin{aligned} a(u_h^*, v) + b(v, \lambda^*) &= f(v) \quad \forall v \in \widetilde{W}^h(\Omega), \\ b(u_h^*, \phi) &= 0 \quad \forall \phi \in \widetilde{V}^h(\Gamma). \end{aligned} \tag{4}$$

Any vector w corresponding to the function $w \in \widetilde{W}^h(\Omega)$ (note that we are using the same symbol for the function and its vector representation) can be decomposed as follows,

$$w = (w^{(i)}, w^{(c)}, w^{(r)}),$$

where $w^{(i)}$ is the vector with dofs associated with the CR nodes of the subdomain interior, $w^{(c)}$ is the vector with dofs associated with the CR nodes of $\mathscr{V}^{CR}$, and $w^{(r)}$ is the vector with dofs associated with the remaining dofs.

Analogously, let $W \subset \widetilde{W}^h(\Omega)$ be the space corresponding to the vectors with the dofs associated with Γ, then we can decompose any vector w of $w \in W$ as $w = (w^{(c)}, w^{(r)})$.

Now let $W_r = \{w^{(r)} : w \in \widetilde{W}^h(\Omega)\}$, in other words, W_r is the space of functions representing the dofs associated with the CR nodes on Γ, not belonging to the set $\mathscr{V}^{CR}$.

Note that $w^{(r)} \in W_r$ has two degrees of freedom associated with each midpoint on $\Gamma \setminus \mathscr{V}^{CR}$, for instance, if $m \in \Gamma_{kl,h}^{CR}$ then its associated two degrees of freedom are $w_k(m)$ and $w_l(m)$.

We introduce A as a block diagonal matrix with local stiffness matrices as the blocks, i.e., $A := \mathrm{diag}(A_k)_{k=1}^N$ with A_k being the stiffness matrix generated by $a_{k,h}(\cdot,\cdot)$ in the standard nodal basis of $W^h(\Omega_k)$.

Let $B = \mathrm{diag}(B^{(kl)})_{\Gamma_{kl}}$ be a block diagonal matrix with $B^{(kl)}$ related to the edge $\Gamma_{kl} \subset \Gamma$ (for $k > l$) containing only zeros, ones and minus ones as matrix entries, and w_h^* is the vector representation of the function $w_h^* \in W$ (denoted by the same symbol).

We note that each block A_j associated with an inner subdomain Ω_j (subdomain not having an edge on $\partial\Omega$), is singular and therefore cannot be inverted. As part of our FETI-DP algorithm, we enforce continuity at the CR nodes close to the cross-points, i.e., at the CR nodes of $\mathscr{V}^{CR}$, thereby remove the problem of noninvertibility.

We introduce the Schur complement matrix, S, of A, with respect to the unknowns associated with Γ, which is obtained after eliminating the unknowns associated with the subdomain interior. We note that S is a block diagonal matrix.

3 FETI-DP problem

Let $\tilde{A}$ be the matrix obtained from block diagonal matrix A by taking into account the continuity of the degrees of freedom at $\mathscr{V}^{CR}$. Let $\tilde{A}$ be partitioned into

$$\tilde{A} = \begin{pmatrix} A_{ii} & A_{ic} & A_{ir} \\ A_{ci} & A_{cc} & A_{cr} \\ A_{ri} & A_{rc} & A_{rr} \end{pmatrix},$$

where the subscript i and superscript (i) refer to the dofs associated with CR nodes in the subdomain interior, the subscript c and superscript (c) to the dofs associated with the crosspoints, and the subscript r and superscript (r) to the dofs associated with the remaining CR nodes.

The matrix formulation of (4) takes the following form,

$$\begin{pmatrix} A_{ii} & A_{ic} & A_{ir} & 0 \\ A_{ci} & A_{cc} & A_{cr} & 0 \\ A_{ri} & A_{rc} & A_{rr} & (B^{(r)})^T \\ 0 & 0 & B^{(r)} & 0 \end{pmatrix} \begin{pmatrix} u^{(i)} \\ u^{(c)} \\ u^{(r)} \\ \lambda^* \end{pmatrix} = \begin{pmatrix} f_i \\ f_c \\ f_r \\ 0 \end{pmatrix}, \tag{5}$$

where $B^{(r)}$ is the submatrix of B, associated with the CR nodes that are on Γ but not in $\mathcal{V}^{CR}$.

Eliminating the unknowns corresponding to the subdomain interior CR nodes and the crosspoints, i.e., $u^{(i)}$ and $u^{(c)}$, in (5) we arrive at

$$\begin{array}{rcl} \tilde{S}u^{(r)} + (B^{(r)})^T\lambda^* &=& \tilde{f}_r, \\ B^{(r)}u^{(r)} &=& 0, \end{array} \tag{6}$$

where $\tilde{S} = A_{rr} - \begin{pmatrix} A_{ri} & A_{rc} \end{pmatrix} \begin{pmatrix} A_{ii} & A_{ic} \\ A_{ci} & A_{cc} \end{pmatrix}^{-1} \begin{pmatrix} A_{ir} \\ A_{cr} \end{pmatrix}$.

Further eliminating $u^{(r)}$, we obtain the following FETI-DP problem: find $\lambda^* \in M$ such that

$$F(\lambda^*) = d, \tag{7}$$

where $d := -B^{(r)}\tilde{S}^{-1}\tilde{f}_r$ and $F := B^{(r)}\tilde{S}^{-1}(B^{(r)})^T$.

4 Parallel preconditioner

The general idea of our Neumann-Dirichlet preconditioner for the FETI-DP system comes from [14], where the case of nonmatching grids and standard continuous P_1 finite element were considered.

We start by further decomposing the vector $w^{(r)}$ into its two component vectors, i.e.,

$$w^{(r)} = \left(w_\Gamma^{(r)}, w_\Delta^{(r)}\right)^T,$$

where $w_\Gamma^{(r)} = (w_{kl,\Gamma}^{(r)})_{\Gamma_{kl}}$ with

$$w_{kl,\Gamma}^{(r)}(m) = \begin{cases} w_k^{(r)}(m) & if\ \rho_k > \rho_l \\ w_k^{(r)}(m) & if\ \rho_k = \rho_l, \quad k > l, \\ w_l^{(r)}(m) & otherwise \end{cases} \qquad m \in \Gamma_{kl,h}^{CR}$$

i.e., $w_{kl,\Gamma}^{(r)}$ is the vector with those entries of $w^{(r)}$ which are related to Γ_{kl} and to the subdomain Ω_s with the larger coefficient ρ_s, $s = k,l$. In case of equality we pick the ones related to Ω_k with $k > l$. The vector $w_\Delta^{(r)}$ corresponds to the remaining dofs of $w^{(r)}$. Correspondingly, we introduce $W_\Delta = \{w_\Delta^{(r)} : w^{(r)} \in W_r\}$, which is a subspace of W_r, consisting of functions which are defined by the values at the CR nodes on the interface Γ_{kl} belonging to the subdomain Ω_s, $s = k,l$, with the smaller coefficient. We note that $\dim \widetilde{V}^h(\Gamma) = \dim W_\Delta$, which equals the number of CR nodes on $\Gamma \setminus \mathcal{V}^{CR}$.

Let S_Δ be the matrix obtained by restricting the block diagonal Schur complement matrix $S : W \to W$ to W_Δ. Note that this matrix can be represented as a block diagonal matrix with nonsingular diagonal blocks $S_{k,\Delta}$, i.e.

$$S_\Delta := \mathrm{diag}(S_{k,\Delta})_k,$$

where the subscript k runs over the subdomains Ω_k such that $S_{k,\Delta}$ correspond to the CR nodes of $\partial\Omega_k^{CR}$ and these CR nodes which are dofs of $w \in W_\Delta$.

We define the nonsingular block diagonal matrix $B_\Delta : W_\Delta \to W_\Delta$, as

$$B_\Delta := \mathrm{diag}(B^{(r)}_{\Delta,\Gamma_{kl}})_{\Gamma_{kl}\subset\Gamma},$$

where $B^{(r)}_{\Delta,\Gamma_{kl}}$ is a diagonal block of the matrix $B^{(r)}$, corresponding to Γ_{kl} and these CR nodes which are dofs of $w \in W_\Delta$. Note that these blocks are nonsingular.

The parallel preconditioner is then as follows,

$$\mathscr{M}_{DN}^{-1} := B_\Delta^{-T} S_\Delta B_\Delta^{-1},$$

which is nonsingular, and its inverse is $\mathscr{M}_{DN} := B_\Delta S_\Delta^{-1} B_\Delta^T$.

5 Condition number bounds

The main result of this paper is the following theorem which yields a bound for the condition number of the preconditioned system.

Theorem 1 (Condition number estimate). *It holds that*

$$\langle \mathscr{M}_{DN}\lambda,\lambda\rangle \le \langle F\lambda,\lambda\rangle \le C\left(1+\log\left(\frac{H}{h}\right)\right)^2 \langle \mathscr{M}_{DN}\lambda,\lambda\rangle \qquad \forall\lambda\in M,$$

where $H = \max_k \mathrm{diam}(\Omega_k)$ and C is a positive constant independent of the coefficients, and the mesh size parameters H and h. Here $\langle\cdot,\cdot\rangle$ is the standard l_2 inner product.

As a direct consequence of this theorem, we see that the condition number of the preconditioned matrix $\mathscr{M}_{DN}^{-1}F$ is bounded by $C\left(1+\log\left(\frac{H}{h}\right)\right)^2$.

The lower bound in the theorem is obtained by a purely algebraic argument, while we get the upper bound by using several technical results of which the most important one is the estimate of special trace norms of jumps of tangential and normal traces over the interface $\Gamma_{kl} \subset \Gamma$.

References

1. Arbogast, T., Chen, Z.: On the implementation of mixed methods as nonconforming methods for second order elliptic problems. Math. Comp. **64**, 943–972 (1995)
2. Arnold, D.N., Brezzi, F.: Mixed and nonconforming finite element methods: Implementation, postprocessing and error estimates. RAIRO Modél. Math. Anal. Numér. **19**, 7–32 (1985)

3. Brenner, S., Sung, L.: Linear finite element methods for planar linear elasticity. Math. Comp. **59**, 321–338 (1992)
4. Brenner, S.C.: A multigrid algorithm for the lowest-order Raviart-Thomas mixed triangular finite element method. SIAM J. Numer. Anal. **29**, 647–678 (1992)
5. Brenner, S.C.: Two-level additive Schwarz preconditioners for nonconforming finite element methods. Math. Comp. **65**(215), 897–921 (1996)
6. Brenner, S.C., Scott, L.R.: The mathematical theory of finite element methods, *Texts in Applied Mathematics*, vol. 15, third edn. Springer, New York (2008)
7. Brenner, S.C., Sung, L.Y.: Balancing domain decomposition for nonconforming plate elements. Numer. Math. **83**(1), 25–52 (1999)
8. Burman, E., Hansbo, P.: Stabilized Crouzeix-Raviart element for the Darcy-Stokes problem. Numer. Methods Partial Differential Equations **21**(5), 986–997 (2005). DOI 10.1002/num.20076. URL http://dx.doi.org/10.1002/num.20076
9. Chatzipantelidis, P.: A finite volume method based on the Crouzeix-Raviart element for elliptic pde's in two dimensions. Numer. Math. **82**, 409–432 (1999)
10. Cowsar, L.C.: Domain decomposition methods for nonconforming finite element spaces of Lagrange type. Tech. Report TR 93-11, Department of Mathematical Sciences, Rice University, Houston (1993)
11. Crouzeix, M., Raviart, P.A.: Conforming and nonconforming finite elements for solving the stationary Stokes equations I. RAIRO Modél. Math. Anal. Numér. **7**(R-3), 33–76 (1973)
12. Farhat, C., Lesoinne, M., Pierson, K.: A scalable dual-primal domain decomposition method. Numer. Linear Algebra Appl. **7**(7-8), 687–714 (2000). Preconditioning techniques for large sparse matrix problems in industrial applications (Minneapolis, MN, 1999)
13. Hoppe, R.H.W., Wohlmuth, B.: Adaptive multilevel iterative techniques for nonconforming finite element discretizations. East-West J. Numer. Math. **3**, 179–197 (1995)
14. Kim, H.H., Lee, C.O.: A preconditioner for the FETI-DP formulation with mortar methods in two dimensions. SIAM J. Numer. Anal. **42**(5), 2159–2175 (2005)
15. Klawonn, A., Widlund, O.B., Dryja, M.: Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. SIAM J. Numer. Anal. **40**(1), 159–179 (2002)
16. Lazarov, R.D., Margenov, S.D.: On a two-level parallel MIC(0) preconditioning of Crouzeix-Raviart non-conforming FEM systems. In: Numerical methods and applications, *Lecture Notes in Comput. Sci.*, vol. 2542, pp. 192–201. Springer, Berlin (2003). DOI 10.1007/3-540-36487-0_21. URL http://dx.doi.org/10.1007/3-540-36487-0_21
17. Mandel, J., Tezaur, R., Farhat, C.: A scalable substructuring method by Lagrange multipliers for plate bending problems. SIAM J. Numer. Anal. **36**(5), 1370–1391 (1999)
18. Marcinkowski, L.: The mortar element method with locally nonconforming elements. BIT **39**(4), 716–739 (1999)
19. Marcinkowski, L.: Additive Schwarz Method for mortar discretization of elliptic problems with P1 nonconforming finite element. BIT **45**(2), 375–394 (2005)
20. Marcinkowski, L., Rahman, T.: Neumann-Neumann algorithms for a mortar Crouzeix-Raviart element for 2nd order elliptic problems. BIT **48**(3), 607–626 (2008)
21. Rahman, T., Xu, X., Hoppe, R.: Additive Schwarz methods for the Crouzeix-Raviart mortar finite element for elliptic problems with discontinuous coefficients. Numer. Math. **101**(3), 551–572 (2003)
22. Sarkis, M.: Nonstandard coarse spaces and Schwarz methods for elliptic problems with discontinuous coefficients using non-conforming elements. Numer. Math. **77**(3), 383–406 (1997)
23. Thomasset, F.: Implementation of Finite Element Methods for Navier-Stokes Equations. Springer Verlag, New York (1981)
24. Toselli, A., Widlund, O.: Domain decomposition methods—algorithms and theory, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin (2005)

An Adaptive Parallel-in-Time Method with application to a membrane problem

Noha Makhoul Karam[1], Nabil Nassif[2], and Jocelyne Erhel[3]

1 Introduction

Assuming global existence on $[0,\infty)$ (and uniqueness) for a solution to the initial value problem: $$(\mathscr{S}) \quad \begin{cases} \frac{dY}{dt} = F(t,Y), & 0 < t \leq T < \infty, \\ Y(0) = Y_0, \end{cases}$$ we seek in this paper, computing its solution $Y : [0,\infty] \to \mathbb{R}^k$ using a parallel-in-time method, for a given function $F : \mathbb{R} \times \mathbb{R}^k \to \mathbb{R}^k$.

There is no natural parallelism across time since the solution on a time level must be known before the computation of the solution at subsequent time levels can start. However, it could be possible to compute simultaneously on many time levels by providing a multi-processor architecture some initial guesses for the solution at later time levels. Such time-parallel computations may be superposed to parallelism in space variables whenever $(\mathscr{S})$ results from a semi-discretization of a time-dependent partial differential equation. Several parallel-in-time algorithms have been proposed to tackle $(\mathscr{S})$. One of the first has been suggested by Nievergelt [12] in 1964 and led to *multiple shooting methods* of which many variants were developed [2], [3], ... In the eighties and nineties, parabolic multigrid methods and multigrid waveform relaxation have been devised. In 2001, Lions, Maday & Turinici proposed in [5] the *parareal algorithm* that marked a turning point: since its introduction, it was subject to many contributions ([6], [4], [1], ...), in particular during Domain Decomposition Conferences. All those methods are based on the principle of combining coarse and fine resolutions in time, starting with the choice of a most often *regular coarse grid* for the time domain, followed by prediction of starting seed values at the lower ends of the coarse grid intervals, then iteratively proceed with parallel computations on a fine grid within each time-interval yielding updated values at their upper ends. Evaluation of the resulting gaps between predicted and updated values on the coarse grid provides corrections for new seed values. An iterative process is thus pursued until convergence occurs.

In this work, we give a parallel-in-time method that has been first introduced in [10] and experimented on a reaction-diffusion problem having a bounded solution. Two main features are used in this method: (i) the use of an end-of-slice function, strongly related to the behavior of the solution, that permits the automatic generation of a non-uniform coarse grid; (ii) rescaling, within each of the generated slices, the time and the solution variables thus obtaining a sequence of rescaled initial value problems with uniformity properties. Such approach has been used (in its two com-

[1]Université Saint Joseph, Beyrouth e-mail: noha.makhoulkaram@usj.edu.lb .[2]American University of Beirut e-mail: nn12@aub.edu.lb .[3] INRIA, Rennes e-mail: Jocelyne.Erhel@inria.fr

ponents) in [8] and [11] for getting sequentially accurate solutions for stiff and explosive systems and has been exploited in [10] for parallel time integration of several types of initial value problems. The resulting parallel in time integration is done without numerical integration over the coarse grid as it is the case in the parareal method: instead, a concept of similarity between the rescaled systems allows the prediction of starting values at the onset of future slices. We refine here the similarity concepts in order to tackle more problems (having non-bounded solutions) and to increase the accuracy of the predictions thus enhancing speed-ups.
After giving, in section 2, an overview of the automatic coarse grid generation, we define in section 3 some similarity properties that yield a prediction model which is at the core of the adaptive parallel-in-time (APTI) algorithm presented in section 4. Numerical results on a membrane problem are then given in section 5.

2 Automatic Coarse Grid generation

The basic principle of the method is in breaking $(\mathscr{S})$ into a sequence of **shooting values problems**. Specifically, we assume the existence of a shooting-value function $E : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ that permits the initiation of a recurrence process, starting with a **first slice** of the coarse grid, obtained by seeking $\left\{T_1, \left\{Y(t) \in \mathbb{R}^k, 0 \le t \le T_1\right\}\right\}$ such that:

$$(\mathscr{S}_1) \qquad \begin{cases} \frac{dY}{dt} = F(t,Y),\ 0 < t < T_1, \\ Y(0) = Y_0, \\ E(Y(t),Y_0) \neq 0, 0 < t < T_1,\ E(Y(T_1),Y_0) = 0. \end{cases}$$

$Y_1 = Y(T_1)$ becomes the initial condition for a 2^{nd} slice of the coarse grid. More generally, we let for $n > 1$, $Y_{n-1} = Y(T_{n-1})$ and define the system on the n^{th} slice:

$$(\mathscr{S}_n) \qquad \begin{cases} \frac{dY}{dt} = F(t,Y),\ T_{n-1} < t < T_n, \\ Y(T_{n-1}) = Y_{n-1}, \\ E(Y(t),Y_{n-1}) \neq 0,\ T_{n-1} < t < T_n,\ E(Y(T_n),Y_{n-1}) = 0. \end{cases}$$

Based on the *End-Of-Slice (EOS) function* $E(.,.)$, one gets **the coarse grid**:

$$\{0 = T_0 < T_1 < ... < T_n < ...T_{N-1} < T \le T_N\},$$

with the corresponding sequence of starting values of the solution:

$$\{Y_n = Y(T_n) | n = 0, 1, ..., N\}.$$

Two cases of existence of a function $E(.,.)$ have so far been identified ([7]).

a. Case of Explosive solutions
Let $||.|| = ||.||_{\infty,\mathbb{R}^k}$ and assume $\lim_{\mathbf{t}\to\infty} \|\mathbf{Y}(\mathbf{t})\| = \infty$. In that case, given $U, W \in \mathbb{R}^k$, and $D(W) \in \mathbb{R}^{k\times k}$ an invertible matrix depending on W with $||(D^{-1}(W))(V)|| \ge c(W)||V||$, we then let for $S > 0$: $E(U,W) = ||D^{-1}(W)(U-W)|| - S$.
When applied to $(\mathscr{S}_n)$, such function $E(.,.)$ determines the size of the n^{th} slice $[T_{n-1}, T_n]$ by:

$$-S \le E(Y(t),Y_{n-1}) < 0,\ T_{n-1} \le t < T_n \text{ and } E(Y(T_n),Y_{n-1}) = 0. \qquad (1)$$

b. Case of Oscillatory Problems
When the behavior of the solution is oscillatory, over a long period of time, in the sense that there exists a two-dimensional plane $\mathscr{P}$ in $\mathbb{R}^k$ on which the projection of the solution's trajectory rotates about a fixed center ω, then a slice is ended when the solution completes a full, or almost full, rotation in that plane about ω.

3 Parallelizing the shooting values problems $\{\mathscr{S}_n\}$

The sequence of shooting values problems $\{(\mathscr{S}_n)|n=1,...,N\}$ can be computed in a parallel way, provided one is able to predict *accurately*, the coarse grid $\{0,T_1,T_2,...,T_N\}$ and the values of the solution $Y(t)$ on that grid, i.e. $\{Y_0,Y_1,Y_2,...,Y_N\}$.

Rescaling and use of local time and solution:
Dealing uniformly with $\{(\mathscr{S}_n)\}$ is then done through a rescaling technique that changes the variables $\{t,Y(t)\}$ on each time-slice $[T_{n-1},T_n]$, into a new pair $\{s,Z_n(s)\}$:

$$\begin{aligned} t &= T_{n-1}+\beta(Y_{n-1})s, && (2.1)\\ Y(t) &= Y_{n-1}+D(Y_{n-1})Z_n(s). && (2.2)\end{aligned} \qquad (2)$$

where $\beta(Y_{n-1})\equiv\beta_n>0$ and $D(Y_{n-1})\equiv D_n\in\mathbb{R}^{k\times k}$ is an invertible matrix. Thus, each $(\mathscr{S}_n)$ is now equivalent to a shooting value problem, whereby one seeks the pair $\{s_n,\{Z_n(s)\in\mathbb{R}^k,0\le s\le s_n\}\}$, such that:

$$(\mathscr{S}'_n)\qquad \begin{cases} \frac{dZ_n}{ds}=G_n(s,Z_n),\ 0<s<s_n\\ Z_n(0)=0,\\ H_n(Z_n(s))\neq 0,\ 0<s<s_n\ H_n(Z_n(s_n))=0, \end{cases}$$

where:
$G_n(s,Z_n)=\beta_n D_n^{-1}F(T_{n-1}+\beta_n s,Y_{n-1}+D_nZ_n)$ and $H_n(Z_n)=E(Y_{n-1}+D_nZ_n,Y_{n-1})$.
Note the following:
- The rescaled time $s=\frac{t-T_{n-1}}{\beta_n}$ and solution $Z_n(s)$ are *set to* 0 *at the beginning of every slice*.
- The functions G_n and H_n depend on the starting values T_{n-1} and Y_{n-1}.
- The solution function $Z_n(.)$ depends on β_n, on each n^{th} slice, in the sense that *different choices of* β_n *lead to different functions* $Z_n(.)$. However, *independently of* β_n *and* D_n, one has the following identities:

$$\forall\beta_n,\qquad \begin{cases} \beta_n s_n=T_n-T_{n-1}, & (3.1)\\ Z_n(s_n)=D_n^{-1}(Y_n-Y_{n-1}). & (3.2)\end{cases} \qquad (3)$$

These identities are at the core of our prediction model, whereas, if the choice of $\beta(Y_{n-1})$ and $D(Y_{n-1})$ are such that the behavior of the pair $\{s_n,Z_n(s_n)\}$ can be accurately predicted, then the coarse grid $\{T_n\}$ and the values $\{Y_n\}$ of $Y(t)$ on that grid can also be obtained from (3).

Similarity concepts:
The change of variables (2) and the consequent rescaled problems $(\mathscr{S}'_n)$ have been originally proposed in [8] and [11] to handle initial value problems $(\mathscr{S})$ which solutions explode in a finite time. As the computation of these problems present a high sensitivity to the sharp variations of the solution on a short time, one way to circumvent this issue is through appropriate choices of $\{\beta_n,D_n,H_n(.)\}$, so that one inherits "uniformity" on the rescaled systems $\{(\mathscr{S}'_n)\}$. This is done by selecting appropriately the rescaling parameter β_n so as to insure uniform boundedness, independently of n, of $\{s_n\}$, $\|Z_n\|$, $\|G_n\|$ and $\|J_{G_n}\|$ (where J_{G_n} is the jacobian of G_n), thus controlling the stiffness of the problem. In that way, placing the same fine solver on each of the $(\mathscr{S}'_n)$, provides a robust algorithm for solving $(\mathscr{S})$, as proved in [9].
Using this approach for parallel in time solving was done first in [10] and more extensively in [7] on the basis of properties satisfied by the pair $\{s_n,Z_n(s_n)\}$.

Definition 1. Invariance: If the rescaling parameters $\{\beta_n, D_n\}$ are such that $\forall n,\ \ G_n(.,.) = G_1(.,.)$ and $H_n(.) = H_1(.)$, then the rescaled systems $(\mathscr{S}'_n)$ are **invariant** and are all equivalent to the shooting Problem $(\mathscr{S}'_1)$.
In that case one has $\forall n,\ Z_n(.) = Z_1(.)$, $s_n = s_1$ and $Z_n(s_n) = Z_1(s_1)$. Invariance is an ideal and rare case: one unique time-slice allows getting the solution on all time-slices through a simple change of variables. A weaker property is given as follows.

Definition 2. Asymptotic similarity: it occurs when the rescaling parameters $\{\beta_n, D_n\}$ are such that $\lim_{n\to\infty}\{s_n, Z_n(s_n)\} = \{s_L, Z_L(s_L)\}$, where $\{s_L, Z_L(s_L)\}$ are obtained from a limit shooting value problem:

$$(\mathscr{S}_L) \qquad \begin{cases} \frac{dZ_L}{ds} = G_L(s, Z_L),\ 0 < s < s_L \\ Z_L(0) = 0, \\ H_L(Z_L(s)) \neq 0,\ 0 < s < s_L\ H_L(Z_L(s_L)) = 0. \end{cases}$$

In this case, the use of (3) for a prediction purpose is possible after a sequential run on a number of slices n_s, at which point one has:

$$\max_{n>n_s}\{\max\{|s_n - s_{n-1}|, ||Z_n(s_n) - Z_{n-1}(s_{n-1})||\}\} \leq tol, \tag{4}$$

where tol is a user's computation tolerance. We finally consider, based on (4), a weak case of similarity, which can be used in spite of the lack of any evidence of invariance or asymptotic similarity.

Definition 3. Numerical Similarity is considered to be reached, whenever, there exists 2 integers, $n_0 \geq 1$ and n_r sufficiently large, such that:

$$\max_{n_0\leq n\leq n_0+n_r}\{\max\{|s_n - s_{n-1}|, ||Z_n(s_n) - Z_{n-1}(s_{n-1})||\}\} \leq tol, \tag{5}$$

In that case, as in (4), one lets $n_s = n_0 + n_r$.

Remark: in the case where all components of Y_n are *distinct from 0*, then (3.2) is equivalent to $Y_n = D_n(e + Z_n(s_n))$, where $e \in \mathbb{R}^k$ is a vector of 1's, and $Z_n(s_n) = D_n^{-1}Y_n - e$ can be expressed in terms of the vector $R_n = D_n^{-1}Y_n = \{\frac{Y_{n,i}}{Y_{n-1,i}}\}$ (***ratio-vector***). The behavior of $\{Z_n(s_n)\}$ is then equivalent to that of $\{R_n\}$.

Data analysis and prediction model:
The similarity properties determine the behavior of the ordered pairs $\{\{s_n, Z_n(s_n)\}\}$ or $\{\{s_n, R_n\}\}$ and allow the prediction of the pairs $\{\{T_n, Y(T_n)\}\}$, *without any integration on the coarse grid*. Hence, *on the basis of Asymptotic or Numerical Similarity*, let n_s be the number of slices on which a sequential run has been conducted with (5) being reached. We seek a prediction data model on the pairs $\{\{s_n, R_n\}|n > n_s\}$. For that purpose, data analysis is carried out on the sequence: $\mathscr{D}^{(0)} = \{\{s_n, R_n\}|n = n_0, ..., n_s\}$. It leads to the model:

$$\{\{s_n, R_n\}|n > n_s\} = Fit(\mathscr{D}^{(0)}), \tag{6}$$

extrapolating best onto next slices. In case of asymptotic similarity, the data model should also take into consideration the convergence of $\{s_n, R_n\}$ to $\{s_L, R_L\}$ (see [7]). Besides, this model allows to get an estimate on N^0, least number of slices such that:

$$\sum_{n=n_s}^{N^0-1} \beta_n s_n < T \leq \sum_{n=n_s}^{N^0} \beta_n s_n. \tag{7}$$

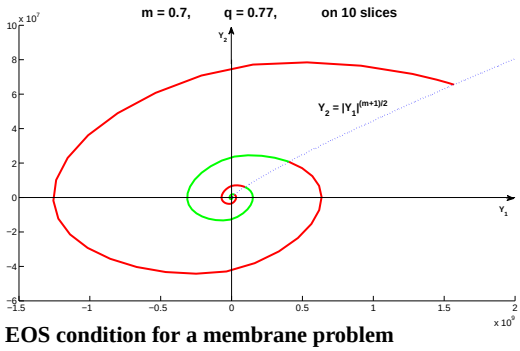
The case of a membrane problem:
Consider the second order IVP where one seeks $y : [0, T] \longrightarrow \mathbb{R}$ ($T \leq \infty$) such that:

$$\begin{cases} y'' - b|y'|^{q-1}y' + |y|^{m-1}y = 0,\ t > 0, & (8.1) \\ y(0) = y_{1,0}, \qquad y'(0) = y_{2,0}. & (8.2) \end{cases} \tag{8}$$

This model describes the motion of a membrane element linked to a spring. When $b>0$, the speed-up of the motion causes a "blow-up" of the solution, case that has been studied by Souplet et al in [13]. In [11], the rescaling method was applied to the case $m>1$ and $q=\frac{2m}{m+1}$ where the solution explodes in finite time. We consider now the case $0<m\leq q\leq\frac{2m}{m+1}\leq 1$. Carrying numerical integration of (8) has shown global existence of the solution on $[0,\infty)$ with **(a)** $\lim_{t\to\infty}|y(t)|=\lim_{t\to\infty}|y^{'}(t)|=\infty$, **(b)** $y(t)$ and $y^{'}(t)$ admit an infinite number of roots in the interval $[0,\infty)$.
Such behavior makes the solution, in the phase-plane (y,y'), spiral outwards about the origin toward infinity. The first step for solving (8) is to write it as a system of first order ODE's. Letting $Y_1(t)=y(t)$ and $Y_2(t)=y'(t)$ makes problem (8) equivalent to an initial value problem of the form $(\mathscr{S})$ where:

$$Y_0=\begin{pmatrix}Y_{1,0}\\ Y_{2,0}\end{pmatrix} \text{ and } Y(t)=\begin{pmatrix}Y_1(t)\\ Y_2(t)\end{pmatrix}, \text{ with } F(t,Y)=F(Y)=\begin{pmatrix}Y_2\\ b|Y_2|^{q-1}\,Y_2\;-\;|Y_1|^{m-1}\,Y_1\end{pmatrix}.$$

Because of the oscillatory behavior of the solution, one possible way to end the n^{th} slice could be whenever the trajectory of the solution, in the Y_1Y_2 phase plane, intersects the curve $Y_2=|Y_1|^{\frac{m+1}{2}}$ in the first quadrant, thus completing an almost full rotation. The oscillating behavior of the solution makes such EOS condition guaranteed to be reached. Thus, one chooses:



EOS condition for a membrane problem

$$\forall\, W=\begin{pmatrix}W_1\\ W_2\end{pmatrix}\in\mathbb{R}^2,\quad H(W)=W_2-|W_1|^{\frac{m+1}{2}},\ \text{and } \beta_n=|Y_{n-1,1}|^{\frac{1-m}{2}}=|Y_{n-1,2}|^{\frac{1-m}{m+1}}.$$

This yields the rescaled systems:

$$\begin{cases}\frac{dZ_{n,1}}{ds}=1+Z_{n,2},\\ \frac{dZ_{n,2}}{ds}=b\gamma_n|1+Z_{n,2}|^{q-1}(1+Z_{n,2})-|1+Z_{n,1}|^{m-1}(1+Z_{n,1}),\ 0<s\leq s_n,\\ Z_{n,1}(0)=Z_{n,2}(0)=0\\ H(Z_n(s))\neq 0,\ 0<s<s_n \text{ and } H(Z_n(s_n))=0,\end{cases}\tag{9}$$

with $\gamma_n=|Y_{n-1,1}|^{\frac{m+1}{2}\left(q-\frac{2m}{m+1}\right)}\leq 1$. Thus, one checks the following [7]:
1. If $q=\frac{2m}{m+1}$, $\forall m\leq 1$, the rescaled systems (9) are invariant and equivalent to finding $Z(s)=(Z_1(s),Z_2(s))$, such that:

$$\begin{cases}\frac{dZ_1}{ds}=1+Z_2,\\ \frac{dZ_2}{ds}=b|1+Z_2|^{q-1}(1+Z_2)-|1+Z_1|^{m-1}(1+Z_1),\ 0<s\leq s_1,\\ Z_1(0)=Z_2(0)=0\\ H(Z(s))\neq 0,\ 0<s<s_1 \text{ and } H(Z(s_1))=0,\end{cases}\tag{10}$$

2. If $0<m\leq q<\frac{2m}{m+1}\leq 1$, then the rescaled systems (9) are asymptotically similar to the limit problem:

$$\begin{cases} \frac{dZ_{L,1}}{ds} = 1 + Z_{L,2}, \\ \frac{dZ_{L,2}}{ds} = -|1 + Z_{L,1}|^{m-1}(1 + Z_{L,1}),\ 0 < s \leq s_L, \\ Z_{L,1}(0) = Z_{L,2}(0) = 0 \\ H(Z_L(s)) \neq 0,\ 0 < s < s_L \text{ and } H(Z_L(s_L)) = 0, \end{cases} \tag{11}$$

4 Adaptive Parallel in Time (APTI) algorithm

The superscripts ^{p} and ^{c} denote predicted and calculated values respectively.
At the core of parallel in time algorithms, one must have a fine solver $\mathscr{F}$ that uniformly handles each of the rescaled problems $(\mathscr{S}'_n)$. It is a software function defined by:
$$(\mathscr{F}) \qquad [Y_n^c, T_n^c] = \mathscr{F}(Y_{n-1}^p, T_{n-1}^p, \beta_n, D_n, F, E, tol),$$
on the basis of the functions F and E, given in $(\mathscr{S}_n)$, with $D_n = D(Y_{n-1})$ and β_n selected to insure obtaining a prediction model on the pairs $\{s_n, Z_n(s_n)\}$; tol is a **global** user's tolerance, the same as that used to check (4) or (5). The function $\mathscr{F}$ is designed so that:

$$\max\left\{\frac{||Y_{n-1} - Y_{n-1}^p||}{||Y_{n-1}||}, \frac{|T_{n-1} - T_{n-1}^p|}{|T_{n-1}|}\right\} = O(tol) \Rightarrow \max\left\{\frac{||Y_n - Y_n^c||}{||Y_n||}, \frac{|T_n - T_n^c|}{|T_n|}\right\} = O(tol). \tag{12}$$

Such fine solver $\mathscr{F}$ is discussed in [9], with a proof of (12) in the case when E is given by $E(U,W) = ||D^{-1}(W)(U-W)|| - S$; $\mathscr{F}$ takes in charge changing $(\mathscr{S}_n)$ to $(\mathscr{S}'_n)$, then uses a high order explicit Runge-Kutta method with a **local** tolerance $tol_1 << tol$ to insure (12).

Theorem 1. *Assuming (12) is satisfied, then:*

$$\begin{cases} \max\left\{\frac{||Y_{n-1} - Y_{n-1}^p||}{||Y_{n-1}||}, \frac{|T_{n-1} - T_{n-1}^p|}{|T_{n-1}|}\right\} = O(tol) \\ \max\left\{\frac{||Y_n^p - Y_n^c||}{||Y_n^p||}, \frac{|T_n^p - T_n^c|}{|T_n^p|}\right\} = O(tol) \end{cases} \Longrightarrow \max\left\{\begin{array}{l} \frac{||Y_n - Y_n^p||}{||Y_n||} \\ \frac{|T_n - T_n^p|}{|T_n|} \end{array}\right\} = O(tol).$$

An **iterative process** can now be initiated using a parallel architecture with P processors. For increasing the speed-up, we adopt a strategy of duplication of sequential tasks on all processors (that reduces communications and avoids idle time).

Initialization step *duplicated on all P processors:*
- Set the iteration index l to 0.
- Solve sequentially problem $\{(\mathscr{S}'_n)\}$ on $m^{(0)} = n_s$ time-slices using $\mathscr{F}$.
- Obtain $\{(T_j^{(0)}, Y_j^{(0)}) | j = 0, ..., m^{(0)}\}$ and let $T^{(0)} = \max\{T_j^{(0)}\}$.
- Compute N^0 according to estimate (7).

Allocation of tasks on the P processors: At this point, the remaining time-slices ($n > m^{(0)}$) are statically allocated, based on a cyclic distribution: processor pr will be assigned slices number n where $(n - m^{(0)})$ is congruent to pr mod P. This provides an optimized synchronization and a *load balanced distribution* of the work.

While $T^{(l)} < T$ (Iterative steps):
(i) All P processors duplicate the task of predicting $\{(T_j^p, Y_j^p) | j = m^{(l)} + 1, ..., N^l\}$, using $Fit(\mathscr{D}^{(l)})$ from (6).
(ii) Every processor $pr \in \{1, ..., P\}$ executes in parallel the following:

a. *pr* solves its first slice n ($n \geq m^{(l)}+1$) and computes Y_n^c and T_n^c using $\mathscr{F}$.
b. Processor *pr* computes $\max\{||Y_n^p - Y_n^c||/||Y_n^p||, |T_n^p - T_n^c|/|T_n^p|\}$.
c. While $\max\{||Y_n^p - Y_n^c||/||Y_n^p||, |T_n^p - T_n^c|/|T_n^p|\} \leq tol$ and $n < N^l$, processor *pr* takes on its assigned next slice, based on theorem 1, and repeats 2(a) (b).
d. If $\max\{||Y_n^p - Y_n^c||/||Y_n^p||, |T_n^p - T_n^c|/|T_n^p|\} > tol$, processor *pr* stops the execution (*the remaining time-slices need not to be solved*). It sends to the master processor (processor 1) the index $\mathscr{I}(pr)$ of the last slice having converged, together with the new $\{T_n^c, Y_n^c\}_{n>m^{(l)}}$.

(iii) Master processor synthesizes the received data and updates the following:
a. Iteration number $l := l+1$ and number of so far solved slices $m^{(l)} := \max_{pr} \mathscr{I}(pr)$.
b. $\left\{\left(T_j^{(l)}, Y_j^{(l)}\right) | j = 0, ..., m^{(l)}\right\}$ with $\left(T_j^{(l)}, Y_j^{(l)}\right) := \left(T_j^{(l-1)}, Y_j^{(l-1)}\right), \forall j = 0, ..., m^{(l-1)}\}$.
c. $T^{*(l)} := \max\{T_j^{(l)}\}$, and $N^{(l)}$ from estimate (7), and the set $D^{(l)}$ to be used by the function *Fit* (as set in (6)).

Then, the master processor sends $T^{(l)}$, $N^{(l)}$ and $D^{(l)}$ to all other processors.

End While

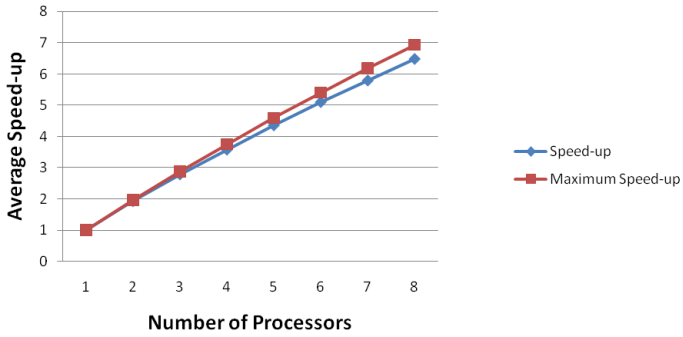
Remark: In case of autonomous problems $F(t,Y) \equiv F(Y)$, one needs not to predict the starting values $\{T_n^p\}$ of the time. Given $\{Y_n^p\}$ only, the rescaling technique allows solving (S'_n) in a local time s, thus providing in parallel $\{s_n\}$ and the size $T_n^c - T_{n-1}^c = \Delta T_n^c = \beta_n s_n$ of time-slices. Then, $\{T_n^c\}$ is reconstituted from received $\{\Delta T_n^c\}$.

5 Numerical results

The table below summarizes some results obtained by the above APTI algorithm on the membrane problem, in the case of asymptotic similarity when $0 < m \leq q < \frac{2m}{m+1}$ and for 8 combinations of the problem parameters m and q, with $b = 1$. The total number of slices N, and therefore the interval of integration $[0,T]$, corresponds to the maximum (or almost) number preventing the explosive solution from exceeding the machine capacity. The total number of iterations vary from one case to another, but in all cases, the results show how small is this number compared to the total number of slices. This ascertains the fast convergence of the method when applied to this type of problems. S_i represents the speed-up obtained when using i processors (compared to the sequential run time of the same rescaling method) and S_i^{max} is the corresponding maximum speed-up stated by Amdhal's law. The following tolerances have been used: $tol = 5 \times 10^{-6}$ (global) and $tol_1 = 10^{-14}$ (local).

Case	1	2	3	4	5	6	7	8
m	**0.8**	**0.7**	**0.7**	**0.6**	**0.6**	**0.6**	**0.5**	**0.5**
q	**0.84**	**0.74**	**0.77**	**0.66**	**0.69**	**0.72**	**0.55**	**0.60**
T	$\approx 10^{14}$	$\approx 10^{29}$	$\approx 10^{28}$	$\approx 10^{14}$	$\approx 10^{18}$	$\approx 10^{30}$	$\approx 10^{17}$	$\approx 10^{25}$
N	**65000**	**65000**	**50000**	**65000**	**65000**	**65000**	**65000**	**65000**
n_s	1499	1143	1471	1156	1414	1993	1053	1385
$\mathbf{n_I}$	**6**	**11**	**12**	**35**	**28**	**23**	**5**	**5**
$\mathbf{S_2}$	**1.88**	**1.93**	**1.93**	**1.96**	**1.94**	**1.91**	**1.96**	**1.94**
S_2^{max}	1.95	1.97	1.94	1.97	1.96	1.94	1.97	1.96
$\mathbf{S_4}$	**3.57**	**3.66**	**3.50**	**3.59**	**3.56**	**3.44**	**3.68**	**3.63**
S_4^{max}	3.74	3.80	3.68	3.80	3.75	3.66	3.81	3.76
$\mathbf{S_8}$	**6.47**	**6.76**	**6.23**	**6.57**	**6.38**	**6.05**	**6.82**	**6.59**
S_8^{max}	6.89	7.12	6.63	7.11	6.94	6.59	7.19	6.96

Actually, the method has been tested on the previous 8 cases, using 2, 3, 4, 5, 6, 7, and 8 processors. The opposite figure shows how the values of speed-up, averaged on the 8 cases above, vary with the number of processors and how close it is to the maximum speed-up.



Conclusion

The application of the adaptive parallel in time algorithm we have presented is not unconditional and requires the prior knowledge of the solution behavior and the existence of an EOS condition inducing the predictability of the end-of-slice values. However, when applicable, APTI algorithm yields a fast convergence due to accurate predictions that do not require any sequential integration on the coarse grid. Besides, not all the remaining time-slices are solved at each iteration and communications are minimized in number and size. Our future work aims at experimenting the method on additional application problems.

References

1. Bal, G., Wu, Q.: Symplectic parareal. In: M. Bercovier, M. Gander, R. Kornhuber, O. Widlund (eds.) DD08, Lecture Notes in computational Science and Eng., pp. 189–202. Springer (2008)
2. Chartier, P., Philippe, B.: A parallel shooting technique for solving dissipative ode's. Computing **51**, 209–236 (1993)
3. Erhel, J., Rault, S.: Algorithme parallèle pour le calcul d'orbites. Techniques et Sciences Informatiques **19**, 649–673 (2000)
4. Farhat, C., Chandesris, M.: Time-decomposed parallel time-integrators. Int. J. Numer. Meth. Engng **58**, 1397–1434 (2003)
5. Lions, J., Maday, Y., Turinici, G.: Résolution d'edp par un schéma en temps "pararéel". C.R.Acad.Sci.Paris **332**, 661–668 (2001)

6. Maday, Y., Bal, G.: A parareal time discretization for non-linear pde's with application to the pricing of an american put. In: I.H. et al (ed.) DD02, Comp. Sc., pp. 189–202. Springer (2002)
7. Makhoul-Karam, N.: Time-slicing, rescaling & ratio-based parallel time integration. TEL (2012). URL http://tel.archives-ouvertes.fr/tel-00743132
8. Nassif, N., Fayad, D., Cortas, M.: Sliced-time computations with rescaling for blowing-up solutions to init. val. pbs. In: S.S. et al. (ed.) ICCS 05, Comp. Sc., pp. 58–65. Springer (2005)
9. Nassif, N., Makhoul-Karam, N., Erhel, J.: Globally adaptive explicit numerical methods for exploding systems of ordinary differential equations. APNUM (2011). URL http://dx.doi.org/10.1016/j.apnum.2011.09.009
10. Nassif, N., Makhoul-Karam, N., Soukiassian, Y.: A new approach for solving evolution problems in time-parallel way. In: V.A. al (ed.) ICCS 06, Comp. Sc., pp. 148–155. Springer (2006)
11. Nassif, N., Makhoul-Karam, N., Soukiassian, Y.: Computation of blowing-up solutions for second-order differential equations using re-scaling techniques. JCAM **227**, 185–195 (2009)
12. Nievergelt, J.: Parallel methods for integration of ode's. Comm. ACM **7**, 731–733 (1964)
13. Souplet, P.: Critical exponents, special large-time behavior and oscillatory blow-up in nonlinear ode's. Differential and Integral Equations **11**, 147–167 (1998)

A Schur Complement Method for DAE Systems in Power System Dynamic Simulations

Petros Aristidou[1], Davide Fabozzi[1], and Thierry Van Cutsem[2]

1 Introduction

Power system dynamic simulations are widely used in industry and academia to provide important information on the dynamic evolution of a system after the occurrence of a disturbance. In modern dynamic simulation software there is the need to represent complex electric equipment that interact with each other directly or through the network. These equipment models represent generators, motors, loads, wind generators, compensators, etc. with all the physics involved and the required controls. This multi-domain modeling leads to large, non-linear, stiff and hybrid (i.e. subject to both continuous and discrete dynamics) Differential and Algebraic Equation (DAE) systems [10].

In these dynamic simulation studies, the speed of simulation is of the utmost importance. The observations resulting from these simulations can be critical in scheduling corrective actions to guard the actual power system against instability. This procedure, called real-time Dynamic Security Assessment, is performed by many power system companies.

Triggered mainly by the developments in parallel processing technologies, some DDMs have already been proposed to speed up simulations. They are mainly based on Schwartz alternating methods and Waveform Relaxation methods [9, 7, 11]. Unlike space domain decomposition, no geometrical information is given to decompose a DAE system [5] and engineers have to rely on a priori information on the system's topology and operation for partitioning. Furthermore, alternating algorithms demand great care in the partitioning of the system and the handling of interface values to ensure the convergence of the methods [1, 8]. If tightly coupled unknowns are mapped to different partitions and an alternating procedure is used, significantly slowed down convergence rates or divergence can be experienced [2].

This paper proposes a robust, accurate and efficient parallel algorithm based on the direct Schur Complement DDM [13]. The algorithm yields significant acceleration when compared to classic, high performance, integrated (applied on the undecomposed system) dynamic simulation algorithms. The two-fold gain comes from utilizing the parallel potential of the method and exploiting the locality and sparsity of power systems. Furthermore, as a direct method, convergence does not depend on the specific partitioning of the system as the interface values are resolved accurately at each step before solving the sub-domain problems. A connection between the

[1] Department of Electrical Engineering and Computer Science, University of Liège, Liège, Belgium, e-mail: p.aristidou@ieee.org .[2] Fund for Scientific Research (FNRS) at the Department of Electrical Engineering and Computer Science, University of Liège, Liège, Belgium, e-mail: t.vancutsem@ulg.ac.be

proposed algorithm and quasi-Newton based integrated algorithms is demonstrated allowing the better comprehension of the algorithm's properties. Finally, an implementation of the algorithm using the shared-memory parallel programming model and some numerical results are presented based on a realistic large-scale test system.

The paper is organized as follows: in Section 2 we present the partitioning scheme of the proposed algorithm. In Section 3, we explain the formulation of the dynamic simulation problem and the solution using the Schur Complement method. In Section 4, some further investigation of the algorithm is made with the help of quasi-Newton integrated algorithms. Implementation specifics and simulation study are reported in Section 5 and followed by closing remarks in Section 6.

2 Power System Modeling

An electric power system, under the phasor approximation [10], can be described in compact form by the following DAE Initial Value Problem:

$$\begin{aligned} 0 &= \boldsymbol{\Psi}(x,V) \\ \boldsymbol{\Gamma}\dot{\mathbf{x}} &= \boldsymbol{\Phi}(x,V) \\ x(t_0) = x_0, V(t_0) &= V_0 \end{aligned} \tag{1}$$

where V is the vector of voltages through the network, x is the expanded state vector containing the differential and algebraic variables (except the voltages) of the system and $\boldsymbol{\Gamma}$ is a diagonal matrix with $\boldsymbol{\Gamma}_{\ell\ell} = \begin{cases} 0, & \text{if the } \ell\text{-th equation is algebraic} \\ 1, & \text{if the } \ell\text{-th equation is differential.} \end{cases}$

The first part of system (1) corresponds to the purely algebraic network equations. The second part describes the remaining DAEs of the system. Discrete events (caused by digital controllers, load tap changing devices, etc.) can alter the power system equations during the simulation. The handling of these discrete events is not presented in this paper [4].

2.1 Power System Partitioning

First, the purely algebraic equations describing the electric network are separated to create one sub-domain. Then, each model of a component connected to the network (such as a synchronous machine, a load, a motor or even a low-voltage distribution network) is separated to form the remaining sub-domains. All the aforementioned devices connected to the network will be called *injectors*. This term encompasses devices that either produce or consume power in normal operating conditions. Each injector is assumed to be connected to a single bus of the network and the interface is on the physical junction between the sub-domains. Extension to two or more connection buses is straightforward [4]. The decomposition is visualized in Fig. 1.

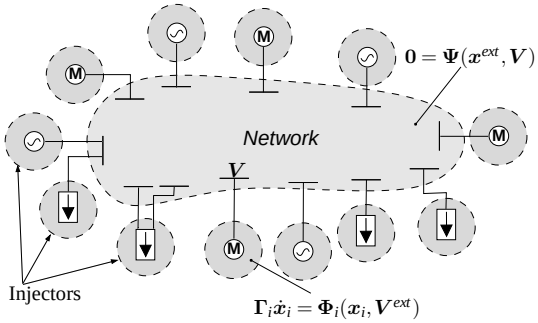


Fig. 1 Decomposed Power System: The Power System is decomposed into the Network and the injectors connected to it. This reveals a star shaped decomposition layout with the Network sub-domain connected to all other sub-domains.

The network sub-domain is described by the algebraic equation system (2) while the sub-domain of each injector i is described by the DAE system (3).

$$\begin{aligned} 0 &= \boldsymbol{\Psi}(x^{ext}, V) \\ x^{ext}(t_0) &= x_0^{ext}, V(t_0) = V_0 \end{aligned} \tag{2}$$

$$\begin{aligned} \boldsymbol{\Gamma}_i \dot{x}_i &= \boldsymbol{\Phi}_i(x_i, V^{ext}) \\ x_i(t_0) &= x_{i0}, V^{ext}(t_0) = V_0^{ext} \end{aligned} \tag{3}$$

Sub-domains numbered $1, \ldots, M-1$ relate to injectors and M relates to the network. Vectors x_i and matrices $\boldsymbol{\Gamma}_i$ are the projections of x and $\boldsymbol{\Gamma}$, defined in (1), on the i sub-domain. The variables of each sub-domain are separated into interior (*int*) variables appearing only in equations of the sub-domain itself and interface (*ext*) variables appearing in equations of both the Network and an injector sub-domain. Thus, for injectors $x_i = [x_i^{int}\ x_i^{ext}]$ and for the Network $V = [V^{int}\ V^{ext}]$ (see Fig. 1).

3 DDM-based Algorithm

3.1 Local System Formulation

Each injector DAE sub-system is algebraized and the resulting non-linear systems of equations are solved with a quasi-Newton method. The local linear systems involved in the solution take on the form of (4) for the injectors and (5) for the network.

$$\underbrace{\begin{bmatrix} A_{1i} & A_{2i} \\ A_{3i} & A_{4i} \end{bmatrix}}_{A_i} \underbrace{\begin{bmatrix} \triangle x_i^{int} \\ \triangle x_i^{ext} \end{bmatrix}}_{\triangle x_i} + \underbrace{\begin{bmatrix} 0 & B_i \end{bmatrix}}_{\widetilde{B}_i} \begin{bmatrix} 0 \\ \triangle V^{ext} \end{bmatrix} = \underbrace{\begin{bmatrix} f_i^{int}(x_i^{int}, x_i^{ext}) \\ f_i^{ext}(x_i^{int}, x_i^{ext}, V^{ext}) \end{bmatrix}}_{f_i} \tag{4}$$

$$\underbrace{\begin{bmatrix} D_1 & D_2 \\ D_3 & D_4 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} \triangle V^{int} \\ \triangle V^{ext} \end{bmatrix}}_{\triangle V} + \Sigma_{j=1}^{M-1} \underbrace{\begin{bmatrix} 0 & C_j \end{bmatrix}}_{\widetilde{C}_j} \begin{bmatrix} 0 \\ \triangle x_j^{ext} \end{bmatrix} = \underbrace{\begin{bmatrix} g^{int}(V^{int}, V^{ext}) \\ g^{ext}(V^{int}, V^{ext}, x^{ext}) \end{bmatrix}}_{g} \tag{5}$$

where A_{1i} (resp. D_1) represents the coupling between interior variables. A_{4i} (resp. D_4) represents the coupling between local interface variables. A_{2i} and A_{3i} (resp. D_2 and D_3) represent the coupling between the local interface and the interior variables and, B_i (resp. C_j) represent the coupling between the local interface variables and the external interface variables of the adjacent sub-domains.

3.2 Global Reduced System Formulation

To formulate the global reduced system involving only the interface variables, the interior variables of the injector sub-domains are eliminated from (4), which yields for the *i*-th injector:

$$S_i \triangle x_i^{ext} + B_i \triangle V^{ext} = \tilde{f}_i \tag{6}$$

where $S_i = A_{4i} - A_{3i} A_{1i}^{-1} A_{2i}$ is the *local* Schur complement matrix and $\tilde{f}_i = f_i^{ext} - A_{3i} A_{1i}^{-1} f_i^{int}$ the corresponding adjusted mismatch values.

Contrary to matrices A_i, which are small but dense and general, matrix D is large but sparse and structurally symmetric. Thus, eliminating the interior variables from (5) would destroy its sparsity and symmetry. Therefore, all the variables of the network sub-domain are included in the reduced system (7).

$$\begin{bmatrix} S_1 & 0 & 0 & \cdots & 0 & B_1 \\ 0 & S_2 & 0 & \cdots & 0 & B_2 \\ 0 & 0 & S_3 & \cdots & 0 & B_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & D_1 & D_2 \\ C_1 & C_2 & C_3 & \cdots & D_3 & D_4 \end{bmatrix} \begin{bmatrix} \triangle x_1^{ext} \\ \triangle x_2^{ext} \\ \triangle x_3^{ext} \\ \vdots \\ \triangle V^{int} \\ \triangle V^{ext} \end{bmatrix} = \begin{bmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \tilde{f}_3 \\ \vdots \\ g^{int} \\ g^{ext} \end{bmatrix} \tag{7}$$

Due to the star layout of the decomposed system (see Fig. 1), the resulting global Schur complement matrix in (7) is block bordered diagonal. Manipulating this structure we can further eliminate all the interface variables of the injector sub-domains and keep only the variables associated to the network sub-domain, as shown in (8).

The elimination factors $C_i S_i^{-1} B_i$ affect only non-zero elements of sub-matrix D_4 thus retaining the original sparsity pattern. This system is solved efficiently using a sparse linear solver to update V at each Newton iteration. Then, the network interface variables (V^{ext}) are backward substituted and the injector sub-domain variables (x_i) are updated independently and in parallel using (4).

$$\underbrace{\begin{bmatrix} D_1 & D_2 \\ D_3 & D_4 - \sum_{i=1}^{N-1} C_i S_i^{-1} B_i \end{bmatrix}}_{\widetilde{D}} \underbrace{\begin{bmatrix} \Delta V^{int} \\ \Delta V^{ext} \end{bmatrix}}_{\Delta V} = \underbrace{\begin{bmatrix} g^{int} \\ g^{ext} - \sum_{i=1}^{M-1} C_i S_i^{-1} \tilde{f}_i \end{bmatrix}}_{\widetilde{g}} \tag{8}$$

3.3 Exploiting Locality

The procedure can be further accelerated by exploiting the locality of the sub-domains. Some sub-domains, described by strongly non-linear systems or with fast changing variables, converge slower. Other sub-domains, with "low dynamic activity", converge faster. This can be exploited in two ways.

First, subdomains with low dynamic activity are detected by measuring the effort (number of Newton iterations) needed for convergence at each discrete time. A subdomain's system is updated if that effort increases above a threshold. Second, a subdomain is declared converged (and stops being solved within the discrete time) if the absolute maximum normalized correction of a Newton solution of the subdomain system becomes smaller than a selected tolerance. Since the low dynamics are detected numerically during the simulation and the tolerance is chosen small enough so as not to disturb the Newton solution, the accuracy of the solution is preserved. Figure 2 shows the full parallel algorithm.

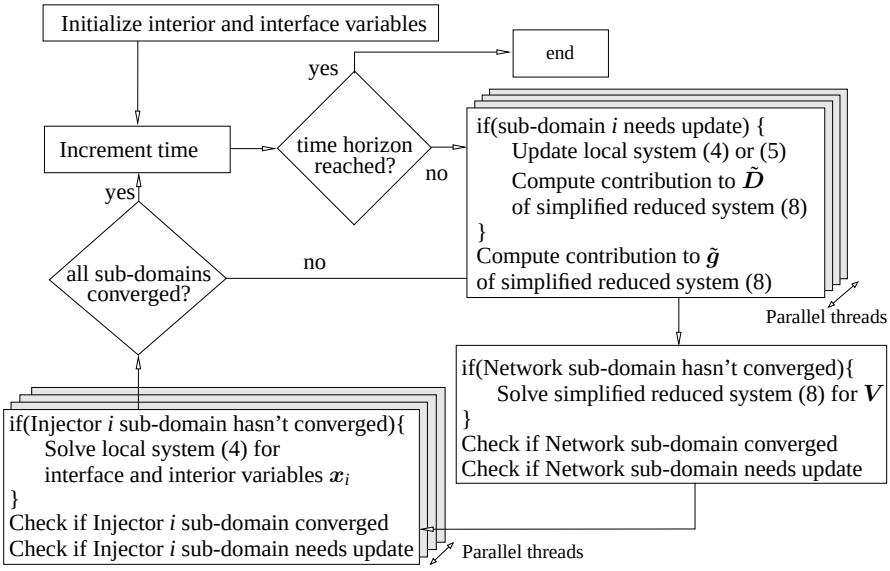


Fig. 2 Parallel Algorithm (P)

4 Further Analysis of the Algorithm

To better understand its properties, Algorithm (P) in Fig. 2 can reformulated into an equivalent quasi-Newton undecomposed scheme with the k-th iteration described:

$$\underbrace{\begin{bmatrix} A_1^{k_1} & 0 & \cdots & 0 & \widetilde{B}_1^{k_1} \\ 0 & A_2^{k_2} & \cdots & 0 & \widetilde{B}_2^{k_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & A_{M-1}^{k_{M-1}} & \widetilde{B}_{M-1}^{k_{M-1}} \\ \widetilde{C}_1^{k_M} & \widetilde{C}_2^{k_M} & \cdots & \widetilde{\mathbf{C}}_{M-1}^{k_M} & D^{k_M} \end{bmatrix}^k}_{\tilde{J}^k} \underbrace{\begin{bmatrix} \triangle x_1 \\ \triangle x_2 \\ \vdots \\ \triangle x_{M-1} \\ \triangle V \end{bmatrix}^k}_{\triangle y^k} = - \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{M-1} \\ g \end{bmatrix}^k}_{F^k} + \underbrace{\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{M-1} \\ r_M \end{bmatrix}^k}_{r^k}$$

$$y^{k+1} = y^k + \triangle y^k$$

where $0 \leq k_j \leq k$ $(j = 1, ..., M)$ and $r_i = \begin{cases} f_i, & \text{if } i\text{-th sub-domain has converged} \\ 0, & \text{otherwise.} \end{cases}$

The approximate Jacobian $\tilde{J}^k$ is used by the method at each iteration k. Every block line i of $\tilde{J}^k$ corresponds to a sub-domain and is updated independently based on sub-domain update criteria [4]. Thus, some block lines can be kept constant for several iterations or even time-steps ($k_i \leq k$).

Furthermore, sub-domains considered to have converged are not solved any more (see Fig. 2). In the equivalent quasi-Newton integrated scheme this corresponds to explicitly setting the mismatch of those sub-domains to zero by introducing some inaccuracy to the method through the correction term r^k. The inaccuracy is bounded and controlled to avoid affecting the accuracy of the final solution.

Using this formulation for Algorithm (P) allows us to utilize a general and well developed framework within which quasi-Newton schemes involving inaccuracy can be described and analyzed [12, 3].

5 Implementation and Numerical Results

The Schur Complement-based DDM was implemented in the simulation software RAMSES, developed at the University of Liège. The benchmark Algorithm (I) is a quasi-Newton scheme applied to the undecomposed DAE system (1). It uses an approximate Jacobian which is updated and factorized if the system hasn't converged after three Newton iterations at any discrete time instant. This method (also referred to as Very Dischonest Newton Method) is considered to be one of the fastest *sequential* algorithms and many traditional industry software use it.

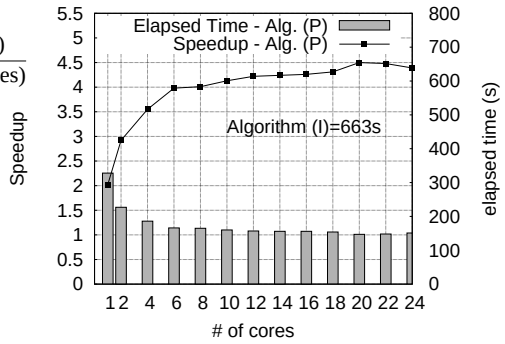
A large-scale model, representative of the Western European main transmission grid, is used. It includes 15226 buses, 21765 branches and 3483 synchronous ma-

chines represented in detail together with their excitation systems, voltage regulators, power system stabilizers, speed governors and turbines. Additionally, 7211 models are included involving induction motors, dynamically modeled loads and equivalents of distribution systems. The resulting, undecomposed, DAE system has 146239 states. The disturbance simulated consists of a short circuit near a bus lasting 5 cycles (100 ms at 50 Hz), that is cleared by opening a double-circuit line. The system is then simulated over a period of 240 s with a time step of 1 cycle (20 ms).

Fig. 3 Speedup index:

$$\frac{\text{time elapsed sequential algorithm (I)}}{\text{time elapsed parallel algorithm (M cores)}}$$

This index shows how faster is the parallel implementation when compared to the fast sequential integrated Algorithm (I) on the same computer.



The same models, algebraization method (second-order Backward Differentiation Formula) and way of handling the discrete events are used in both algorithms. For the solution of the sparse linear systems, HSL MA41 [6] is used and for the dense injector linear systems of Algorithm (P), Intel MKL LAPACK library. The computer used for the simulation is a 24-core, shared memory, AMD Opteron Interlagos (CPU 6238 @ 2.60GHz) running Debian Linux.

Fig. 4 Real-time index:

$$n = \frac{\text{simulation elapsed time}}{\text{simulated physical time}}$$

This index shows how faster was the simulation than the simulated time. This is an important index for control center applications where the speed of computation is an issue for operator decision.

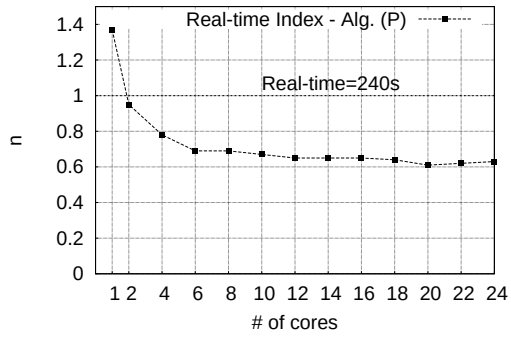


Figure 3 shows that the DDM-based algorithm is already twice faster than the benchmark in sequential execution. This speedup is mainly attributed to the exploitation of locality in the decomposed algorithm (Section 3.3). As we proceed to parallel execution, the proposed algorithm performs up to 4.5 times faster. Figure 4 shows the real-time potential of the algorithm in parallel execution.

The ratio of the interface system to the subdomain systems is very important to the performance of the algorithm since it corresponds to the ratio between the sequential portion of the code and the parallel portion of the code. A higher ratio leads to better speedup and avoids the saturation observed when increasing the number of cores. The size of the interface system (8) is the same as of the network subdomain (5), that is approx. 30 000 for the test-system considered. At the same time, the subdomain systems include approx. 120 000 states. Thus, the size ratio is approx. 4, which explains why a relatively small speedup is observed after 6 cores and the speedup saturates at 4.5 times.

6 Conclusion

In this paper a Schur Complement-based algorithm for dynamic simulation of electric power systems has been outlined. The algorithm yields acceleration of the simulation procedure in two ways. On the one hand, the procedure is accelerated numerically, by exploiting the locality of the sub-domain systems and avoiding many unnecessary computations (factorizations, evaluations, solutions). On the other hand, the procedure is accelerated computationally, by exploiting the parallelization opportunities inherent to DDMs.

References

1. Crow, M., Ilic, M., White, J.: Convergence properties of the waveform relaxation method as applied to electric power systems. In: Circuits and Systems, 1989., IEEE International Symposium on, pp. 1863 –1866 vol.3 (1989)
2. CRSA, RTE, TE, TU/e: D4.1: Algorithmic requirements for simulation of large network extreme scenarios. Tech. rep. URL `http://www.fp7-pegase.eu/`
3. Dennis, J., Walker, H.: Inaccuracy in quasi-Newton methods: Local improvement theorems. Mathematical Programming at Oberwolfach II **22**, 70–85 (1984)
4. Fabozzi, D.: Decomposition, Localization and Time-Averaging Approaches in Large-Scale Power System Dynamic Simulation. Ph.D. thesis, University of Liège (2012)
5. Guibert, D., Tromeur-Dervout, D.: A Schur Complement Method for DAE/ODE Systems in Multi-Domain Mechanical Design. Domain Decomposition Methods in Science and Engineering XVII pp. 535–541 (2008)
6. HSL(2011): A collection of Fortran codes for large scale scientific computation. URL `http://www.hsl.rl.ac.uk`
7. Ilic'-Spong, M., Crow, M.L., Pai, M.A.: Transient Stability Simulation by Waveform Relaxation Methods. Power Systems, IEEE Transactions on **2**(4), 943–949 (1987)
8. Jackiewicz, Z., Kwapisz, M.: Convergence of waveform relaxation methods for differential-algebraic systems. SIAM Journal on Numerical Analysis **33**(6), 2303–2317 (1996)
9. Kron, G.: Diakoptics: the piecewise solution of large-scale systems. MacDonald (1963)
10. Kundur, P.: Power system stability and control. McGraw-hill New York (1994)
11. La Scala, M., Bose, A., Tylavsky, D., Chai, J.: A highly parallel method for transient stability analysis. Power Systems, IEEE Transactions on **5**(4), 1439 –1446 (1990)
12. Ortega, J., Rheinboldt, W.: Iterative Solution of Nonlinear Equations in Several Variables. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (1987)

13. Saad, Y.: Iterative methods for sparse linear systems, second edn. Society for Industrial and Applied Mathematics (2003)

FETI solvers for non-standard finite element equations based on boundary integral operators

Clemens Hofreither[1], Ulrich Langer[2], and Clemens Pechstein[2]

1 Introduction

This paper is devoted to the construction and analysis of Finite Element Tearing and Interconnecting (FETI) methods for solving large-scale systems of linear algebraic equations arising from a new non-standard finite element discretization of the diffusion equation. This discretization technique uses PDE-harmonic trial functions in every element of a polyhedral mesh. The generation of the local stiffness matrices utilizes boundary element techniques. For these reasons, this non-standard finite element method can also be called a BEM-based FEM or Trefftz-FEM.

The FETI method was introduced by Farhat and Roux in [1] and has been generalized and analyzed by many people, see, e.g., [11] and [7] for the corresponding references. The Boundary Element Tearing and Interconnecting (BETI) method was later introduced by Langer and Steinbach [6] as the boundary element counterpart of the FETI method. The analysis of the convergence of the BETI method is heavily based on the spectral equivalences between FEM- and BEM-approximated Steklov-Poincaré operators. Similar techniques are used for the analysis of the BEM-based FETI methods considered in this paper. Due to space constraints, this analysis is however postponed to a forthcoming article. In the present work, we derive the solver, state the convergence results without proof, and present numerical results.

2 A skeletal variational formulation

Let $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, be a bounded Lipschitz domain, and let us consider the following diffusion problem in the standard weak form: find $u \in H^1(\Omega)$ such that u matches the given Dirichlet data g_D on Γ_D and satisfies the variational equation

$$\int_\Omega \alpha \nabla u \cdot \nabla v \, dx = \int_\Omega f v \, dx + \int_{\Gamma_N} g_N v \, ds \; \forall v \in H^1_D(\Omega) = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\} \quad (1)$$

where α is the uniformly positive and bounded diffusion coefficient, f is a given forcing term, $\Gamma_D \subseteq \partial\Omega$ is the Dirichlet boundary with positive surface measure, $\Gamma_N = \partial\Omega \setminus \overline{\Gamma_D}$ is the Neumann boundary with prescribed conormal derivative g_N.

[1] Doctoral Program "Computational Mathematics" Johannes Kepler University, Linz, Austria, e-mail: clemens.hofreither@dk-compmath.jku.at .[2] Institute of Computational Mathematics, Johannes Kepler University, Linz, Austria, e-mail: {ulanger}(clemens.pechstein)@numa.uni-linz.ac.at

Consider a decomposition $\mathscr{T}$ of the domain Ω into polytopal elements $T \in \mathscr{T}$. In contrast to a standard FEM method, we allow the mesh to consist of a mixture of rather general polygons (in 2d) or polyhedra (in 3d). We now require that the coefficient function α is piecewise constant with respect to $\mathscr{T}$, i.e., $\alpha|_T(x) \equiv \alpha_T \; \forall T \in \mathscr{T}$.

On every element T, we introduce the local harmonic extension operator $\mathscr{H}_T : H^{1/2}(\partial T) \to H^1(T)$ which maps any $g_T \in H^{1/2}(\partial T)$ to the unique weak solution $u_T \in H^1(T)$ of the local PDE $-\operatorname{div}(\alpha_T \nabla u_T) = 0$ with Dirichlet boundary condition $u_T|_{\partial T} = g_T$. Furthermore, we define the local *Steklov-Poincaré operator* $S_T : H^{1/2}(\partial T) \to H^{-1/2}(\partial T)$ by $S_T u_T = \gamma^1 \mathscr{H}_T u_T$, where γ^1 is the conormal derivative operator which takes the form $\gamma^1 = n \cdot \alpha \nabla$ for sufficiently regular arguments.

If we introduce the *skeleton* $\Gamma_S := \bigcup_{T \in \mathscr{T}} \partial T$ and denote by $H^{1/2}(\Gamma_S)$ the trace space of $H^1(\Omega)$-functions onto the skeleton, we can formulate the skeletal variational problem: find $u \in H^{1/2}(\Gamma_S)$ with $u|_{\Gamma_D} = g_D$ such that

$$a(u,v) = \langle F, v \rangle \quad \forall v \in \mathscr{W}_D = \{v \in \mathscr{W} = H^{1/2}(\Gamma_S) : v|_{\Gamma_D} = 0\}, \tag{2}$$

where the bilinear from $a(u,v)$ and the linear form $\langle F, v \rangle$ are defined by $a(u,v) = \sum_{T \in \mathscr{T}} \langle S_T u|_{\partial T}, v|_{\partial T} \rangle$ and $\langle F, v \rangle = \sum_{T \in \mathscr{T}} \left[\int_T f \mathscr{H}_T(v|_{\partial T})\, dx + \int_{\partial T \cap \Gamma_N} g_N v\, ds \right]$, respectively. It is easy to see that the skeletal variational formulation (2) is equivalent to the standard variational formulation (1) in the sense that the solution of the former is the skeletal trace of the solution of the latter [3].

3 Approximation of the Steklov-Poincaré operator

It is well-known [10] that the Steklov-Poincaré operator S_T can be expressed as

$$S_T = \alpha_T (V_T^{-1}(\tfrac{1}{2} I + K_T)) = \alpha_T (D_T + (\tfrac{1}{2} I + K_T') V_T^{-1} (\tfrac{1}{2} I + K_T))$$

in terms of the boundary integral operators defined on every element boundary ∂T,

$$\begin{aligned} &V_T : H^{-1/2}(\partial T) \to H^{1/2}(\partial T), \quad K_T : H^{1/2}(\partial T) \to H^{1/2}(\partial T), \\ &K_T' : H^{-1/2}(\partial T) \to H^{-1/2}(\partial T), \quad D_T : H^{1/2}(\partial T) \to H^{-1/2}(\partial T), \end{aligned}$$

called, in turn, the *single layer potential*, *double layer potential*, *adjoint double layer potential*, and *hypersingular* operators. They are defined by means of the fundamental solution of the Laplace equation.

We construct a computable approximation as follows. We assume that each element boundary ∂T has a shape-regular mesh $\mathscr{F}_T$ which consists of line segments in $\mathbb{R}^2$ and of triangles in $\mathbb{R}^3$, and that these local meshes match across elements. On this mesh, we construct a space $\mathscr{Z}_T^h$ of piecewise constant functions and define, given $u \in H^{1/2}(\partial T)$, the discrete variable $w_T^h \in \mathscr{Z}_T^h$ by solving the discrete variational problem $\langle V_T w_T^h, z_T^h \rangle = \langle (\frac{1}{2} I + K_T) u, z_T^h \rangle$ for all $z_T^h \in \mathscr{Z}_T^h$. A computable approxima-

tion to S_T is then given by $\widetilde{S}_T u := \alpha_T(D_T u + (\frac{1}{2}I + K_T')w_T^h)$. The approximation $\widetilde{S}_T$ remains self-adjoint and its kernel is given by the constant functions, just as for S_T. Furthermore, it satisfies the spectral equivalence

$$\tilde{c}_T \langle S_T v, v\rangle \le \langle \widetilde{S}_T v, v\rangle \le \langle S_T v, v\rangle \qquad \forall v \in H^{1/2}(\partial T) \tag{3}$$

with $\tilde{c}_T \in (0, \frac{1}{4}]$. Replacing, in (2), S_T by its approximations $\widetilde{S}_T$, we obtain the inexact skeletal variational formulation: find $u \in H^{1/2}(\Gamma_S)$ with $u|_{\Gamma_D} = g_D$ such that

$$\widetilde{a}(u,v) := \sum_{T\in\mathscr{T}} \langle \widetilde{S}_T u|_{\partial T}, v|_{\partial T}\rangle = \langle F, v\rangle \qquad \forall v \in \mathscr{W}_D.$$

The positive constant $\tilde{c}_T$ in (3) depends on the geometry of the element T. For robust error estimates, it is necessary to bound $\tilde{c}_T$ from below uniformly for all elements. Recently, explicit bounds for these constants have been obtained, starting with a paper by Pechstein [8] which relied on the Jones parameter and a constant in an isoperimetric inequality. These results were employed in the rigorous *a priori* error analysis of the BEM-based FEM [3, 2] and have later been simplified in [4].

Theorem 1 ([4]). *Let $\Omega \subset \mathbb{R}^3$. Assume that there exists a shape-regular simplicial mesh $\Xi(\Omega')$ of an open, bounded superset $\Omega' \supset \overline{\Omega}$ of Ω such that each element $T \in \mathscr{T}$ is a union of simplices from $\Xi(\Omega')$, and the number of simplices per element T is uniformly bounded. Furthermore, assume that the boundary meshes $\mathscr{F}_T$, $T \in \mathscr{T}$, are shape-regular.*

Then, the contraction constants $\tilde{c}_T$, $T \in \mathscr{T}$, are uniformly bounded away from 0 *in terms of the mesh regularity parameters.*

4 Discretization

By assumption, $\mathscr{F} := \bigcup_{T\in\mathscr{T}} \mathscr{F}_T$ describes a shape-regular triangulation of the skeleton Γ_S. On this mesh, we construct the discrete trial space $\mathscr{W}^h \subset H^{1/2}(\Gamma_S)$ of piecewise linear, continuous functions on the skeleton and set $\mathscr{W}_D^h := \mathscr{W}^h \cap \mathscr{W}_D$. After this discretization, we aim to find $u^h \in \mathscr{W}^h$ with $u^h|_{\Gamma_D} = g_D$ such that

$$\widetilde{a}(u^h, v^h) = \langle F, v^h\rangle \qquad \forall v^h \in \mathscr{W}_D^h. \tag{4}$$

Rigorous error estimates of optimal order for this discretized variational problem can be found in [3, 2]. Equivalently, (4) can be written as an operator equation

$$Au^h = F \tag{5}$$

with $A : \mathscr{W}^h \to (\mathscr{W}_D^h)^*$. The associated stiffness matrix in the canonical nodal basis shares many properties with the stiffness matrix obtained from a standard finite element method like sparsity, symmetry and positive definiteness.

5 A FETI solver

In the following, we derive a solution method for (5) based on the ideas of the FETI substructuring approach, originally proposed by Farhat and Roux [1]. Our derivation closely follows that of the classical FETI method. Thus, we refer to the monographs [11] and [7] and the references therein for further details and proofs.

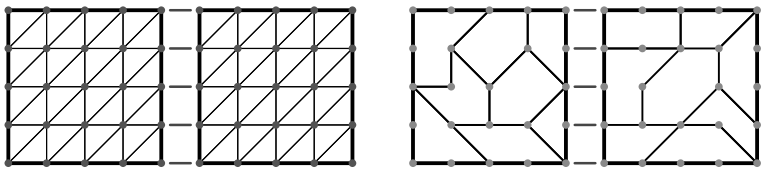


Fig. 1 Sketch of domain decomposition approach in 2D for a rectangular domain with $N = 2$ subdomains. *Left:* FETI substructuring. *Right:* FETI-like substructuring for the BEM-based FEM.

We decompose Ω into non-overlapping subdomains $(\Omega_i)_{i=1}^N$ in agreement with the polyhedral mesh $\mathscr{T}$, that is, $\overline{\Omega}_i = \bigcup_{T\in\mathscr{T}_i} \overline{T}$ with an associate decomposition $(\mathscr{T}_i)_{i=1}^N$. We set $H_i := \operatorname{diam}\Omega_i$ and $H := \max_{i=1}^N H_i$. Every subdomain Ω_i has an associated skeleton $\bigcup_{T\in\mathscr{T}_i} \partial T$ and discrete skeletal trial spaces $\mathscr{W}^h(\Omega_i)$ and $\mathscr{W}_D^h(\Omega_i)$, constructed as in Section 4. In the following, we assume that the problem has been homogenized with respect to the given Dirichlet data g_D, such that $u^h \in \mathscr{W}_D^h$.

Both the operator A and the functional F in (5) can be written as a sum of local contributions $A_i : \mathscr{W}^h(\Omega_i) \to \mathscr{W}^h(\Omega_i)^*$ and $f_i \in \mathscr{W}^h(\Omega_i)^*$ such that $\sum_{i=1}^N A_i(u|_{\Omega_i}) = \sum_{i=1}^N f_i$, where here and in the sequel we drop the superscript h since all functions are discrete from now on. Indeed, all relevant functions live in spaces of piecewise linear functions which have natural nodal bases. Therefore, we will not distinguish in the following between functions and the coefficient vectors representing them with respect to the nodal basis, nor between operators and their matrix representations.

We introduce the Schur complement $\widetilde{S}_i = A_{i,\Gamma\Gamma} - A_{i,\Gamma I}A_{i,II}^{-1}A_{i,I\Gamma}$ of the subdomain stiffness matrix A_i. The blocks $A_{i,\Gamma\Gamma}, A_{i,\Gamma I}, A_{i,I\Gamma}, A_{i,II}$ are chosen such that the subscripts Γ and I correspond to the boundary and inner degrees of freedom, i.e.,

$$A_i w = \begin{bmatrix} A_{i,\Gamma\Gamma} & A_{i,\Gamma I} \\ A_{i,I\Gamma} & A_{i,II} \end{bmatrix} \begin{bmatrix} w_\Gamma \\ w_I \end{bmatrix}.$$

Eliminating the interior unknowns in (5) yields the equivalent minimization problem

$$u = \arg\min_{v\in\mathscr{W}_D^h(\Gamma_S^H)} \frac{1}{2}\sum_{i=1}^N \langle \widetilde{S}_i v|_{\partial\Omega_i}, v|_{\partial\Omega_i}\rangle - \sum_{i=1}^N \langle g_i, v|_{\partial\Omega_i}\rangle, \tag{6}$$

where $\Gamma_S^H = \bigcup_{i=1}^N \partial\Omega_i$ is the coarse skeleton, $\mathscr{W}_D^h(\Gamma_S^H)$ is the trace space of discrete functions $\mathscr{W}_D^h(\Omega)$ onto Γ_S^H, and g_i is a suitably adjusted forcing term.

Let $\mathscr{W}^h(\partial\Omega_i) := \{v|_{\partial\Omega_i} : v \in \mathscr{W}^h(\Omega_i)\}$ denote a space of discrete boundary functions. We then introduce the broken space $Y := \prod_{i=1}^N Y_i$ with $Y_i := \{v \in \mathscr{W}^h(\partial\Omega_i) : v|_{\Gamma_D} = 0\}$. In order to enforce continuity of the functions in Y, we introduce the jump operator $B : Y \to \mathbb{R}^{N_\Lambda}$, where $N_\Lambda \in \mathbb{N}$ is the total number of constraints. Here we assume fully redundant constraints, i.e., for every node on a subdomain interface, constraints corresponding to all neighboring subdomains are introduced. This choice implies that B is not surjective, and we define the space of Lagrange multipliers as the range $\Lambda := \operatorname{Range} B \subseteq \mathbb{R}^{N_\Lambda}$ and consider B as a mapping $Y \to \Lambda$.

Using the jump operator, we rewrite (6) as $u = \arg\min_{y \in \ker B} \frac{1}{2}\sum_{i=1}^N \langle \widetilde{S}_i y_i, y_i\rangle - \sum_{i=1}^N \langle g_i, y_i\rangle$. Introducing Lagrange multipliers to enforce the constraint $By = 0$, we obtain the saddle point formulation

$$\begin{bmatrix} \widetilde{S} & B^\top \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} g \\ 0 \end{bmatrix}, \tag{7}$$

for $u \in Y$ and $\lambda \in \Lambda$, with the block matrices and vectors $\widetilde{S} = \operatorname{diag}(\widetilde{S}_1, \ldots, \widetilde{S}_N)$, $B = (B_1, \ldots, B_N)$, $u = (u_1, \ldots, u_N)^\top$, $g = (g_1, \ldots, g_N)^\top$. From (7), we see that the local skeletal functions u_i satisfy the relationship

$$\widetilde{S}_i u_i = g_i - B_i^\top \lambda. \tag{8}$$

For a *non-floating* domain Ω_i, that is, one that shares a part of the Dirichlet boundary such that $\partial\Omega_i \cap \Gamma_D \neq \emptyset$, $\widetilde{S}_i$ is positive definite and thus invertible. For a *floating* domain Ω_i, the kernel of $\widetilde{S}_i$ consists only of the constant functions, and we parameterize it by the operator $R_i : \mathbb{R} \to \ker \widetilde{S}_i \subset Y_i$ which maps a scalar to the corresponding constant function. Under the condition that the right-hand side is orthogonal to the kernel, i.e.,

$$\langle g_i - B_i^\top \lambda, R_i \zeta\rangle = 0 \qquad \forall \zeta \in \mathbb{R}, \tag{9}$$

the local problem (8) is solvable and we have $u_i = \widetilde{S}_i^\dagger (g_i - B_i^\top \lambda) + R_i \xi_i$ with some $\xi_i \in \mathbb{R}$. Here, $\widetilde{S}_i^\dagger$ denotes a pseudo-inverse of $\widetilde{S}_i$. For non-floating domains Ω_i, we set $\widetilde{S}_i^\dagger = \widetilde{S}_i^{-1}$.

We set $Z := \prod_{i=1}^N \mathbb{R}^{\dim(\ker \widetilde{S}_i)}$ and introduce the operator $R : Z \to Y$ by $(R\xi)|_{\Omega_i} := R_i \xi_i$ for floating Ω_i and $(R\xi)|_{\Omega_i} := 0$ for non-floating Ω_i. The local solutions u can then be expressed by

$$u = \widetilde{S}^\dagger (g - B^\top \lambda) + R\xi \tag{10}$$

under the compatibility condition $R^\top B^\top \lambda = R^\top g$ derived from (9). Inserting (10) into the second line of (7) yields $B\widetilde{S}^\dagger g - B\widetilde{S}^\dagger B^\top \lambda + BR\xi = 0$, and together with the compatibility condition and using the notations $F = B\widetilde{S}^\dagger B^\top$ and $G = BR$, we obtain the dual saddle point problem

$$\begin{bmatrix} F & -G \\ G^\top & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \xi \end{bmatrix} = \begin{bmatrix} B\widetilde{S}^\dagger g \\ R^\top g \end{bmatrix}. \tag{11}$$

With a self-adjoint operator $Q : \Lambda \to \Lambda$ which is assumed to be positive definite on the range of G and which will be specified later, we define the projector $P = I - QG(G^\top QG)^{-1}G^\top$ from Λ onto the subspace $\Lambda_0 := \ker G^\top \subset \Lambda$ of admissible increments. The choice $\lambda_g := QG(G^\top QG)^{-1}R^\top g \in \Lambda$ ensures that $G^\top \lambda_g = R^\top g$, and thus, with $\lambda = \lambda_0 + \lambda_g$, we can homogenize (11) such that we only search for a $\lambda_0 \in \Lambda_0$ with

$$F\lambda_0 - G\xi = B\widetilde{S}^\dagger g - F\lambda_g. \tag{12}$$

Applying the projector $P^\top$ to this equation and noting that $P^\top G = 0$, we obtain the following formulation of the dual problem: find $\lambda_0 \in \Lambda_0$ such that

$$P^\top F\lambda_0 = P^\top(B\widetilde{S}^\dagger g - F\lambda_g) = P^\top B\widetilde{S}^\dagger(g - B^\top \lambda_g). \tag{13}$$

It can be shown that $P^\top F$ is self-adjoint and positive definite on Λ_0. Thus, the problem (13) has a unique solution which may be computed by CG iteration in the subspace Λ_0. Once $\lambda = \lambda_0 + \lambda_g$ has been computed, we see that applying $(G^\top QG)^{-1}G^\top Q$ to (12) yields $\xi = (G^\top QG)^{-1}G^\top QB\widetilde{S}^\dagger(B^\top \lambda - g)$. The unknowns u_i may then be obtained by solving the local problems (10), and the unknowns in the interior of each Ω_i may be recovered by solving local Dirichlet problems.

Preconditioners for FETI are typically constructed in the form PM^{-1} with a suitable operator $M^{-1} : \Lambda \to \Lambda$. The FETI Dirichlet preconditioner adapted to our setting, is given by the choice $M^{-1} = B\widetilde{S}B^\top$ and works well for constant or mildly varying coefficient α. In this case, the choice $Q = I$ works satisfactorily.

To deal with coefficient jumps, we need to employ a *scaled* or *weighted jump operator* as introduced in [9] and analyzed in [5]. We restrict ourselves to the case of subdomain-wise constant coefficient α, i.e., $\alpha(x) = \alpha_i$ for $x \in \Omega_i$.

Let $x^h \in \partial\Omega_i$ refer to a boundary node. We introduce weighted counting functions δ_j via piecewise linear interpolation on the facets of the coarse skeleton Γ_S^H of the nodal values defined by $\delta_j(x^h) = \alpha_j / (\sum_{k \in \{1,\dots,N\} : x^h \in \partial\Omega_k} \alpha_k)$ for $x^h \in \partial\Omega_j$ and 0 otherwise, $j = 1,\dots,N$. We introduce diagonal scaling matrices $D_i : \Lambda \to \Lambda$, $i = 1,\dots,N$, operating on the space of Lagrange multipliers. Consider two neighboring domains Ω_i and Ω_j sharing a node $x^h \in \partial\Omega_i \cap \partial\Omega_j$. Let $k \in \{1,\dots,N_\Lambda\}$ denote the index of the Lagrange multiplier associated with this node and pair of subdomains. Then, the k-th diagonal entry of D_i is set to $\delta_j(x^h)$, and the k-th diagonal entry of D_j to $\delta_i(x^h)$. Diagonal entries of D_i not associated with a node on $\partial\Omega_i$ are set to 0.

The *weighted jump operator* $B_D : Y \to \Lambda$ is now given by $B_D = [D_1 B_1, \dots, D_N B_N]$, and the weighted Dirichlet preconditioner by $M_D^{-1} = B_D \widetilde{S} B_D^\top$. In this case, a possible choice for Q is simply $Q = M_D^{-1}$. Alternatively, Q can be replaced by a suitable diagonal matrix as described in [5].

6 Convergence Analysis

The convergence analysis proceeds by the idea of spectral equivalences between the BEM-based FEM Schur complements $\widetilde{S}_i$ and the Schur complements which occur

in a standard one-level FETI method, allowing us to transfer the known condition estimates from the FETI literature to our case. This is similar to the approach used in the analysis of the BETI method [6]. For space reason, we cannot give this analysis here, and it must be postponed to a forthcoming paper. Here we only state the main results. Under standard assumptions, we can prove the condition number estimate

$$\kappa(P^\top F|_{\Lambda_0}) \le C(\overline{\alpha}/\underline{\alpha})\max_{i=1,\dots,N}(H_i/h_i)$$

for the non-preconditioned case, where $\overline{\alpha} = \max_{x\in\Omega}\alpha(x)$, $\underline{\alpha} = \min_{x\in\Omega}\alpha(x)$, and the constant C depends only on mesh regularity parameters. For the preconditioned case, with the choice $Q = M_D^{-1}$, we have the condition number estimate

$$\kappa(PM_D^{-1}P^\top F|_{\Lambda_0}) \le C\,(1+\log(\max_{i=1,\dots,N}(H_i/h_i)))^2.$$

7 Numerical experiments

We solve the pure Dirichlet boundary value problem $-\Delta u = 0$ in Ω and $u(x) = -(2\pi)^{-1}\log|x-x^\star|$ on $\partial\Omega$. The 2d domain Ω (Figure 2, left) is discretized by an irregular polygonal mesh. The source point $x^\star = (-1,1)$ lies outside of Ω.

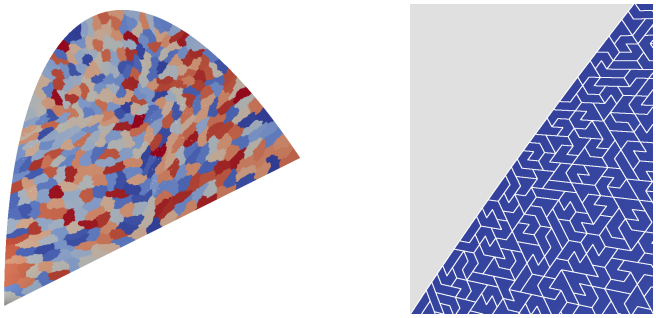


Fig. 2 *Left:* Ω partioned into $N = 400$ subdomains. *Right:* Zoom into the polygonal mesh.

The polygonal mesh $\mathscr{T}$ is constructed by applying the graph partitioner METIS to a standard triangular mesh consisting of 524,288 triangles, resulting in a polygonal mesh with 99,970 elements, most of which are unions of 5 or 6 triangles, cf. Figure 2, right. The domain decomposition $\{\Omega_i\}$ is obtained by applying METIS a second time on top of the mesh $\mathscr{T}$, see Figure 2, left.

We use the Dirichlet preconditioner with multiplicity scaling and a suitable diagonal matrix for Q as described in [5], and solve the dual system by the corresponding PCG iteration. In Table 1, we give the number of CG iterations required to reduce the initial residual by a factor of 10^{-8} without and with Dirichlet preconditioner, and provide some CPU times for varying number N of subdomains.

N	total time	avg. loc. time	#iter	# Lagrange
25	32.23 / **20.49**	0.0776 / **0.0759**	133 / **29**	5875
50	30.19 / **19.10**	0.0317 / **0.0310**	135 / **30**	8962
100	26.64 / **17.70**	0.0135 / **0.0131**	131 / **31**	13012
200	23.69 / **17.41**	0.0059 / **0.0057**	134 / **36**	19056
400	21.06 / **16.13**	0.0027 / **0.0026**	123 / **34**	27324
800	20.23 / **17.68**	0.0013 / **0.0013**	109 / **36**	39304
1600	22.19 / **20.96**	0.0006 / **0.0006**	095 / **35**	56632

Table 1 Results of the non-preconditioned (left) / **preconditioned (right)** CG solver. Columns: number of subdomains, total CPU time for the solution in seconds, averaged time for solving the local problems in seconds, number of iterations, number of Lagrange multipliers.

Acknowledgements The authors gratefully acknowledge the financial support by the Austrian Science Fund (FWF) under the grant DK W1214, project DK4.

References

1. Farhat, C., Roux, F.X.: A method of finite element tearing and interconnecting and its parallel solution algorithm. Int. J. Numer. Meth. Engrg. **32**, 1205–1227 (1991)
2. Hofreither, C.: L_2 error estimates for a nonstandard finite element method on polyhedral meshes. J. Numer. Math. **19**(1), 27–39 (2011)
3. Hofreither, C., Langer, U., Pechstein, C.: Analysis of a non-standard finite element method based on boundary integral operators. Electron. Trans. Numer. Anal. **37**, 413–436 (2010)
4. Hofreither, C., Pechstein, C.: A rigorous error analysis of coupled FEM-BEM problems with arbitrary many subdomains. In: T. Apel, O. Steinbach (eds.) Advanced Finite Element Methods and Applications, pp. 109–130. Springer (2012)
5. Klawonn, A., Widlund, O.B.: FETI and Neumann-Neumann iterative substructuring methods: connections and new results. Comm. Pure Appl. Math. **54**(1), 57–90 (2001)
6. Langer, U., Steinbach, O.: Boundary element tearing and interconnecting method. Computing **71**(3), 205–228 (2003)
7. Pechstein, C.: Finite and Boundary Element Tearing and Interconnecting Solvers for Multiscale Problems. Springer, Heidelberg (2013)
8. Pechstein, C.: Shape-explicit constants for some boundary integral operators. Appl. Anal. **92**(5), 949–974 (2013)
9. Rixen, D., Farhat, C.: A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. Int. J. Numer. Meth. Engrg. **44**(4), 489–516 (1999)
10. Steinbach, O.: Numerical approximation methods for elliptic boundary value problems – Finite and boundary elements. Springer, New York (2008)
11. Toselli, A., Widlund, O.: Domain Decoposition Methods – Algorithms and Theory. Springer, Berlin, Heidelberg (2004)

Domain decomposition methods for problems of unilateral contact between elastic bodies with nonlinear Winkler covers

Ihor I. Prokopyshyn[1], Ivan I. Dyyak[2], Rostyslav M. Martynyak[1], and Ivan A. Prokopyshyn[2]

1 Introduction

Thin covers from different materials are often applied in engineering to improve the functional properties of the surfaces of machines and structures components. On the other hand, thin covers with certain mechanical properties are used to model the real microstructure of surfaces, adhesion and glue bondings [6, 14, 15].

The classical methods for solution of contact problems for bodies with thin covers are grounded on integral equations and are reviewed in work [15]. Nowadays, one of the most effective numerical methods for such contact problems are methods, based on variational formulations and finite element approximations.

Efficient approach for solution of multibody contact problems is the use of domain decomposition methods (DDMs). Many DDMs for contact problems without covers are obtained on discrete level [3, 16]. Among DDMs, proposed on continuous level for contact problems without covers are methods presented in [1, 9, 12]. Domain decomposition methods for solution of problem of ideal contact between two bodies, connected through nonlinear Winkler layer are proposed in [2, 8]. These methods are based on saddle-point formulation and conjugate gradient methods.

In current contribution we consider a problem of unilateral contact between elastic bodies with nonlinear Winkler covers. We give variational formulations of this problem in the form of nonquadratic variational inequality on convex set and nonlinear variational equation in the whole space, and present theorems about existence and uniqueness of their solution. Furthermore, we propose on continuous level a class of parallel domain decomposition methods for solving the nonlinear variational equation, which corresponds to original contact problem. In each iteration of these methods we have to solve in a parallel way linear variational equations in separate bodies, which are equivalent in a weak sense to linear elasticity problems with Robin boundary conditions on possible contact areas. These DDMs are based on abstract nonstationary iterative methods for variational equations in Banach spaces. They are the generalization of domain decomposition methods, proposed by us earlier in [4, 5, 10] for unilateral contact problems without covers. Some particular cases of proposed DDMs can be viewed as a modification of semismooth Newton method [7]. The numerical analysis of obtained DDMs is made for plane contact problems using finite element approximations.

[1]Pidstryhach IAPMM NASU, Naukova 3-b, Lviv, 79060, Ukraine, e-mail: ihor84@gmail.com .[2] Ivan Franko National University of Lviv, Universytetska 1, Lviv, 79000, Ukraine

2 Statement of the problem

Consider a unilateral contact of N elastic bodies $\Omega_\alpha \subset \mathbb{R}^3$ with sufficiently smooth boundaries Γ_α, $\alpha = 1,2,...,N$ (Fig.1a). Suppose that across each contact surface there is a nonlinear Winkler layer. Denote $\Omega = \bigcup_{\alpha=1}^{N} \Omega_\alpha$.

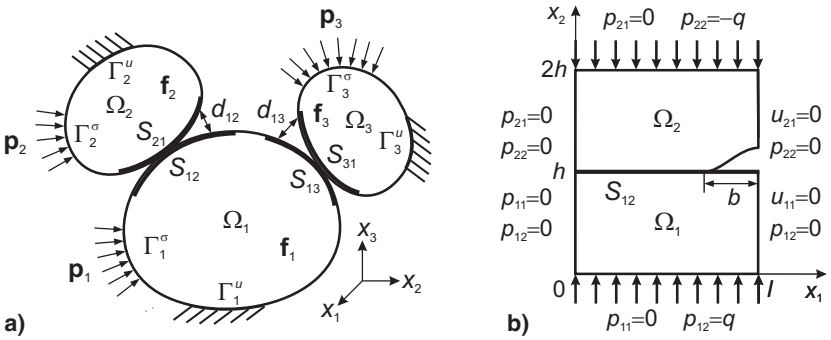


Fig. 1 Unilateral contact between several elastic bodies through nonlinear Winkler layers

A stress-strain state in point $\mathbf{x} = (x_1, x_2, x_3)^\top$ of each solid Ω_α is described by the displacement vector $\mathbf{u}_\alpha = u_{\alpha i}\,\mathbf{e}_i$, the tensor of strains $\hat{\boldsymbol{\varepsilon}}_\alpha = \varepsilon_{\alpha ij}\,\mathbf{e}_i\,\mathbf{e}_j$ and the tensor of stresses $\hat{\boldsymbol{\sigma}}_\alpha = \sigma_{\alpha ij}\,\mathbf{e}_i\,\mathbf{e}_j$. These quantities satisfy the following relations:

$$\sum_{j=1}^{3} \frac{\partial \sigma_{\alpha ij}(\mathbf{x})}{\partial x_j} + f_{\alpha i}(\mathbf{x}) = 0\,,\ \ \mathbf{x} \in \Omega_\alpha\,,\ \ i = 1,2,3\,, \tag{1}$$

$$\sigma_{\alpha ij}(\mathbf{x}) = \sum_{k,l=1}^{3} C_{\alpha ijkl}(\mathbf{x})\,\varepsilon_{\alpha kl}(\mathbf{x})\,,\ \ \varepsilon_{\alpha ij} = \frac{1}{2}\left(\frac{\partial u_{\alpha i}}{\partial x_j} + \frac{\partial u_{\alpha j}}{\partial x_i}\right),\ \ i,j = 1,2,3\,, \tag{2}$$

where $f_{\alpha i}$ are the components of volume forces vector $\mathbf{f}_\alpha = f_{\alpha i}\,\mathbf{e}_i$, and $C_{\alpha ijkl}$ are symmetric elasticity constants, which are bounded in the following sense:

$$(\exists\, b_\alpha, c_\alpha > 0)\ (\forall \mathbf{x})\ \left\{ b_\alpha \sum_{i,j=1}^{3} \varepsilon_{\alpha ij}^2 \le \sum_{i,j,k,l=1}^{3} C_{\alpha ijkl}\,\varepsilon_{\alpha ij}\,\varepsilon_{\alpha kl} \le c_\alpha \sum_{k,l=1}^{3} \varepsilon_{\alpha kl}^2 \right\}. \tag{3}$$

On the boundary Γ_α introduce a local orthonormal coordinate system $\boldsymbol{\xi}_\alpha$, $\boldsymbol{\eta}_\alpha$, $\mathbf{n}_\alpha$, where $\mathbf{n}_\alpha$ is an outer unit normal, and $\boldsymbol{\xi}_\alpha$, $\boldsymbol{\eta}_\alpha$ are unit tangents. Then the vectors of displacements and stresses on Γ_α can be written in the following way: $\mathbf{u}_\alpha = u_{\alpha\xi}\,\boldsymbol{\xi}_\alpha + u_{\alpha\eta}\,\boldsymbol{\eta}_\alpha + u_{\alpha n}\,\mathbf{n}_\alpha$, $\boldsymbol{\sigma}_\alpha = \hat{\boldsymbol{\sigma}}_\alpha \cdot \mathbf{n}_\alpha = \sigma_{\alpha\xi}\,\boldsymbol{\xi}_\alpha + \sigma_{\alpha\eta}\,\boldsymbol{\eta}_\alpha + \sigma_{\alpha n}\,\mathbf{n}_\alpha$.

Suppose, that the boundary Γ_α consists of three disjoint parts: $\Gamma_\alpha = \Gamma_\alpha^u \bigcup \Gamma_\alpha^\sigma \bigcup S_\alpha$, $\Gamma_\alpha^u = \overline{\Gamma_\alpha^u}$, $\Gamma_\alpha^u \neq \emptyset$, $S_\alpha \neq \emptyset$. On the part Γ_α^u homogenous Dirichlet boundary conditions are prescribed, and on the part Γ_α^σ we consider Neumann boundary conditions:

$$\mathbf{u}_\alpha(\mathbf{x}) = 0,\ \mathbf{x} \in \Gamma_\alpha^u;\ \ \boldsymbol{\sigma}_\alpha(\mathbf{x}) = \mathbf{p}_\alpha(\mathbf{x}),\ \mathbf{x} \in \Gamma_\alpha^\sigma\,. \tag{4}$$

The part $S_\alpha = \bigcup_{\beta\in B_\alpha} S_{\alpha\beta}$, $\bigcap_{\beta\in B_\alpha} S_{\alpha\beta} = \emptyset$ is the possible contact area of body Ω_α with the other bodies. Here $S_{\alpha\beta}$ is the possible unilateral contact area of body Ω_α with body Ω_β, and $B_\alpha \subset \{1,2,...,N\}$ is the set of the indices of all bodies in contact with body Ω_α. We assume that the surfaces $S_{\alpha\beta} \subset \Gamma_\alpha$ and $S_{\beta\alpha} \subset \Gamma_\beta$ are sufficiently close ($S_{\alpha\beta} \approx S_{\beta\alpha}$), and $\mathbf{n}_\alpha(\mathbf{x}) \approx -\mathbf{n}_\beta(\mathbf{x}')$, $\mathbf{x} \in S_{\alpha\beta}$, $\mathbf{x}' = P(\mathbf{x}) \in S_{\beta\alpha}$, where $P(\mathbf{x})$ is the projection of point $\mathbf{x}$ on $S_{\alpha\beta}$. Let $d_{\alpha\beta}(\mathbf{x}) = \pm\|\mathbf{x}-\mathbf{x}'\|$ be a distance between bodies Ω_α and Ω_β before the deformation. We suppose that possible contact areas $S_{\alpha\beta}$ and $S_{\beta\alpha}$, $\beta \in B_\alpha$, $\alpha = 1,...,N$ have nonlinear Winkler covers. Total compression $w_{\alpha\beta}$ of these covers is related with normal contact stress as follows: $\sigma_{\alpha n}(\mathbf{x}) = \sigma_{\beta n}(\mathbf{x}') = g_{\alpha\beta}\left(w_{\alpha\beta}(\mathbf{x})\right)$, $\mathbf{x} \in S_{\alpha\beta}$, $\mathbf{x}' \in S_{\beta\alpha}$, where $g_{\alpha\beta}$ is given nonlinear continuous function, which satisfies the following conditions:

$$g_{\alpha\beta}(0) = 0\,, \quad (\forall y,z)\,\left\{ y < z \Rightarrow g_{\alpha\beta}(y) < g_{\alpha\beta}(z)\right\}, \tag{5}$$

$$\left(\exists M_{\alpha\beta} > 0\right)\,(\forall y,z)\,\left\{\left|g_{\alpha\beta}(y) - g_{\alpha\beta}(z)\right| \le M_{\alpha\beta}\,|y-z|\right\}. \tag{6}$$

On possible contact zones $S_{\alpha\beta}$, $\beta \in B_\alpha$, $\alpha = 1,2,...,N$ we consider the following unilateral contact conditions through nonlinear Winkler layers:

$$\sigma_{\alpha\xi}(\mathbf{x}) = \sigma_{\beta\xi}(\mathbf{x}') = 0\,, \;\; \sigma_{\alpha\eta}(\mathbf{x}) = \sigma_{\beta\eta}(\mathbf{x}') = 0\,, \tag{7}$$

$$\sigma_{\alpha n}(\mathbf{x}) = \sigma_{\beta n}(\mathbf{x}') = g_{\alpha\beta}\left(w_{\alpha\beta}(\mathbf{x})\right) \le 0, \;\; u_{\alpha n}(\mathbf{x}) + u_{\beta n}(\mathbf{x}') + w_{\alpha\beta}(\mathbf{x}) \le d_{\alpha\beta}(\mathbf{x}), \tag{8}$$

$$\left[u_{\alpha n}(\mathbf{x}) + u_{\beta n}(\mathbf{x}') + w_{\alpha\beta}(\mathbf{x}) - d_{\alpha\beta}(\mathbf{x})\right]\sigma_{\alpha n}(\mathbf{x}) = 0\,, \;\; \mathbf{x}' = P(\mathbf{x})\,, \;\; \mathbf{x} \in S_{\alpha\beta}\,. \tag{9}$$

3 Variational formulations

For each body Ω_α consider Sobolev space $V_\alpha = [H^1(\Omega_\alpha)]^3$ and the closed subspace $V_\alpha^0 = \{\mathbf{u}_\alpha \in V_\alpha :\; \mathbf{u}_\alpha = 0 \text{ on } \Gamma_\alpha^u\}$. All values of the elements from these spaces on the parts of boundary Γ_α should be understood as traces. The trace of element $\mathbf{u}_\alpha \in V_\alpha$ on the part Γ_α^u should belong to space $[H^{1/2}(\Gamma_\alpha^u)]^3$, and the trace of element from V_α^0 on the part $\Xi_\alpha = \operatorname{int}(\Gamma_\alpha \setminus \Gamma_\alpha^u)$ should belong to $[H_{00}^{1/2}(\Xi_\alpha)]^3$.

Define Hilbert space $V_0 = \prod_{\alpha=1}^N V_\alpha$ with scalar product $(\mathbf{u},\mathbf{v})_{V_0} = \sum_{\alpha=1}^N (\mathbf{u}_\alpha,\mathbf{v}_\alpha)_{V_\alpha}$ and norm $\|\mathbf{u}\|_{V_0} = (\mathbf{u},\mathbf{u})_{V_0}^{1/2}$, $\mathbf{u},\mathbf{v} \in V_0$. Moreover, introduce the following spaces $W = \{\mathbf{w} = (w_{\alpha\beta})_{\{\alpha,\beta\}\in Q}^\top :\; w_{\alpha\beta} \in H_{00}^{1/2}(\Xi_\alpha)\}$ and $U_0 = V_0 \times W = \{\mathbf{U} = (\mathbf{u},\mathbf{w})^\top :\; \mathbf{u} \in V_0, \mathbf{w} \in W\}$, where $Q = \{\{\alpha,\beta\} :\; \alpha \in \{1,2,...,N\}\,,\; \beta \in B_\alpha\}$.

In space U_0 consider the closed convex set of all displacements, which satisfy nonpenentration contact conditions: $K = \{\mathbf{U} \in U_0 :\; u_{\alpha n} + u_{\beta n} + w_{\alpha\beta} \le d_{\alpha\beta} \text{ on } S_{\alpha\beta},\; \{\alpha,\beta\} \in Q\}$, where $u_{\alpha n} = \mathbf{n}_\alpha \cdot \mathbf{u}_\alpha \in H_{00}^{1/2}(\Xi_\alpha)$, $w_{\alpha\beta}, d_{\alpha\beta} \in H_{00}^{1/2}(\Xi_\alpha)$.

Let us introduce bilinear form $A(\mathbf{u},\mathbf{v}) = \sum_{\alpha=1}^N a_\alpha(\mathbf{u}_\alpha,\mathbf{v}_\alpha)$, $\mathbf{u},\mathbf{v} \in V_0$, $a_\alpha(\mathbf{u}_\alpha,\mathbf{v}_\alpha) = \int_{\Omega_\alpha} \hat{\boldsymbol{\sigma}}_\alpha(\mathbf{u}_\alpha) : \hat{\boldsymbol{\varepsilon}}_\alpha(\mathbf{v}_\alpha)\,d\Omega$, such that $A(\mathbf{u},\mathbf{u})$ represents the total elastic deformation energy of the bodies, linear form $L(\mathbf{u}) = \sum_{\alpha=1}^N l_\alpha(\mathbf{u}_\alpha)$, $l_\alpha(\mathbf{u}_\alpha) = \int_{\Omega_\alpha} \mathbf{f}_\alpha \cdot \mathbf{u}_\alpha\,d\Omega +$

$\int_{\Gamma_\alpha^\sigma} \mathbf{p}_\alpha \cdot \mathbf{u}_\alpha \, dS$, $\mathbf{f}_\alpha \in [L_2(\Omega_\alpha)]^3$, $\mathbf{p}_\alpha \in [H_{00}^{-1/2}(\Xi_\alpha)]^3$, which is equal to external forces work, and nonquadratic functional $H(\mathbf{w}) = \sum_{\{\alpha,\beta\}\in Q} \int_{S_{\alpha\beta}} \left[\int_0^{w_{\alpha\beta}} g_{\alpha\beta}(z)\, dz \right] dS$, $\mathbf{w} \in W$, which represents the total deformation energy of nonlinear Winkler layers.

We have shown, that bilinear form A is symmetric, continuous and coercive if condition (3) holds, and nonquadratic functional H is Gateaux differentiable: $H'(\mathbf{w},\mathbf{z}) = \sum_{\{\alpha,\beta\}\in Q} \int_{S_{\alpha\beta}} g_{\alpha\beta}(w_{\alpha\beta})\, z_{\alpha\beta}\, dS$, $\mathbf{w},\mathbf{z} \in W$.

Theorem 1. *Suppose that conditions (3), (5), (6) hold. Then problem (1), (2), (4), (7)–(9) has an alternative weak formulation as the following minimization problem:*

$$F(\mathbf{U}) = A(\mathbf{u},\mathbf{u})/2 - L(\mathbf{u}) + H(\mathbf{w}) \to \min_{\mathbf{U}\in K}. \tag{10}$$

Moreover, there exists a unique solution of problem (10), and this problem is equivalent to the following nonquadratic variational inequality on set K:

$$F'(\mathbf{U}, \mathbf{V}-\mathbf{U}) = A(\mathbf{u}, \mathbf{v}-\mathbf{u}) - L(\mathbf{v}-\mathbf{u}) + H'(\mathbf{w}, \mathbf{z}-\mathbf{w}) \geq 0\,, \ \forall (\mathbf{v},\mathbf{z})^\top \in K\,. \tag{11}$$

Except this variational formulation, we also have proposed another weak formulation of original contact problem in the form of nonlinear variational equation.

Let us introduce the following nonquadratic functional in space V_0:

$$J(\mathbf{u}) = \sum_{\{\alpha,\beta\}\in Q} \int_{S_{\alpha\beta}} \left[\int_0^{d_{\alpha\beta}-u_{\alpha n}-u_{\beta n}} g^-_{\alpha\beta}(z)\, dz \right] dS\,, \ \mathbf{u} \in V_0\,, \tag{12}$$

where $g^-_{\alpha\beta}(z) = \{\,0\,,\, z \geq 0\,\} \vee \{\, g_{\alpha\beta}(z)\,,\, z < 0\,\}$ is nonlinear function.

Functional $J(\mathbf{u})$ is nonnegative and Gateaux differentiable in V_0: $J'(\mathbf{u},\mathbf{v}) = -\sum_{\{\alpha,\beta\}\in Q} \int_{S_{\alpha\beta}} g^-_{\alpha\beta}(d_{\alpha\beta} - u_{\alpha n} - u_{\beta n})\, [v_{\alpha n} + v_{\beta n}]\, dS$. We have shown that if conditions (5) and (6) hold, then Gateaux differential $J'(\mathbf{u},\mathbf{v})$ satisfies the following properties: $(\forall \mathbf{u} \in V_0)\ (\exists \tilde{R} > 0)\ (\forall \mathbf{v} \in V_0)\, \{\, |J'(\mathbf{u},\mathbf{v})| \leq \tilde{R}\|\mathbf{v}\|_{V_0}\}$, $(\exists \tilde{D} > 0)\, (\forall \mathbf{u},\mathbf{v},\mathbf{w} \in V_0)\, \{\, |J'(\mathbf{u}+\mathbf{w},\mathbf{v}) - J'(\mathbf{u},\mathbf{v})\,| \leq \tilde{D}\|\mathbf{v}\|_{V_0}\|\mathbf{w}\|_{V_0}\,\}$, $(\forall \mathbf{u},\mathbf{v} \in V_0)$ $\{\, J'(\mathbf{u}+\mathbf{v},\mathbf{v}) - J'(\mathbf{u},\mathbf{v}) \geq 0\,\}$. These properties helped us to prove the next theorem.

Theorem 2. *Suppose that conditions (3), (5) and (6) hold. Then the contact problem (1), (2), (4), (7)–(9) is equivalent to problem (1), (2), (4), (7) with the following nonlinear boundary value conditions on the possible contact areas:*

$$\sigma_{\alpha n}(\mathbf{x}) = \sigma_{\beta n}(\mathbf{x}') = g^-_{\alpha\beta}\left(d_{\alpha\beta}(\mathbf{x}) - u_{\alpha n}(\mathbf{x}) - u_{\beta n}(\mathbf{x}')\right)\,, \ \mathbf{x}' = P(\mathbf{x})\,, \ \mathbf{x} \in S_{\alpha\beta}\,, \tag{13}$$

and it is equivalent in weak sense to the next nonquadratic minimization problem:

$$F_1(\mathbf{u}) = A(\mathbf{u},\mathbf{u})/2 - L(\mathbf{u}) + J(\mathbf{u}) \to \min_{\mathbf{u}\in V_0}. \tag{14}$$

Moreover, problem (14) has a unique solution and is equivalent to the following nonlinear variational equation in space V_0:

$$F_1'(\mathbf{u},\mathbf{v}) = A(\mathbf{u},\mathbf{v}) + J'(\mathbf{u},\mathbf{v}) - L(\mathbf{v}) = 0, \ \forall \mathbf{v} \in V_0\,, \ \mathbf{u} \in V_0\,. \tag{15}$$

4 Nonstationary iterative methods

In reflexive Banach space V consider an abstract nonlinear variational equation

$$\Phi(\mathbf{u},\mathbf{v}) = Y(\mathbf{v}), \ \forall \mathbf{v} \in V, \ \mathbf{u} \in V, \tag{16}$$

where $\Phi : V \times V \to \mathbb{R}$ is a functional, which is linear in $\mathbf{v}$, but nonlinear in $\mathbf{u}$, and $Y : V \to \mathbb{R}$ is linear continuous form. For numerical solution of (16) consider the following nonstationary iterative method [5, 11]:

$$G^k(\mathbf{u}^{k+1},\mathbf{v}) = G^k(\mathbf{u}^k,\mathbf{v}) - \gamma^k \left[\Phi(\mathbf{u}^k,\mathbf{v}) - Y(\mathbf{v}) \right], \ k = 0,1,\dots, \tag{17}$$

where $G^k : V \times V \to \mathbb{R}$ are some given bilinear forms, $\gamma^k \in \mathbb{R}$ are iterative parameters, and $\mathbf{u}^k \in V$ is the k-th approximation to the exact solution of problem (16).

Theorem 3. [5] *Suppose that functional Φ satisfies the following properties:* $(\forall \mathbf{u} \in V)(\exists R_\Phi > 0)(\forall \mathbf{v} \in V)\{ |\Phi(\mathbf{u},\mathbf{v})| \le R_\Phi \|\mathbf{v}\|_V \}$, $(\exists D_\Phi > 0)$ $(\forall \mathbf{u},\mathbf{v},\mathbf{w} \in V)$ $\{ |\Phi(\mathbf{u}+\mathbf{w},\mathbf{v}) - \Phi(\mathbf{u},\mathbf{v})| \le D_\Phi \|\mathbf{v}\|_V \|\mathbf{w}\|_V \}$, $(\exists B_\Phi > 0)(\forall \mathbf{u},\mathbf{v} \in V)\{ \Phi(\mathbf{u}+\mathbf{v},\mathbf{v}) - \Phi(\mathbf{u},\mathbf{v}) \ge B_\Phi \|\mathbf{v}\|_V^2 \}$. *Then nonlinear variational equation (16) has a unique solution $\bar{\mathbf{u}} \in V$. In addition, suppose that bilinear forms G^k, $k = 0,1,\dots$ are symmetric, continuous with constant $M_G^* > 0$, coercive with constant $B_G^* > 0$, and the following conditions hold:* $(\exists k_0 \in \mathbb{N}_0)(\forall k \ge k_0)$ $(\forall \mathbf{u} \in V)$ $\{ G^k(\mathbf{u},\mathbf{u}) \ge G^{k+1}(\mathbf{u},\mathbf{u}) \}$, $(\exists \varepsilon \in (0, \gamma^*),\ \gamma^* = B_\Phi B_G^* / D_\Phi^2)(\exists k_1)(\forall k \ge k_1)\{ \gamma^k \in [\varepsilon, 2\gamma^* - \varepsilon] \}$. *Then $\|\mathbf{u}^k - \bar{\mathbf{u}}\|_V \underset{k\to\infty}{\to} 0$, where $\{\mathbf{u}^k\} \subset V$ is obtained by iterative method (17).*

5 Domain decomposition schemes

Now let us apply nonstationary iterative method (17) for solving the nonlinear variational equation (15), which corresponds to original contact problem. This equation can be written in form (16), where $\Phi(\mathbf{u},\mathbf{v}) = A(\mathbf{u},\mathbf{v}) + J'(\mathbf{u},\mathbf{v})$, $Y(\mathbf{v}) = L(\mathbf{v})$, $\mathbf{u},\mathbf{v} \in V$, $V = V_0$, and iterative method (17) applied to solve (15) rewrites as follows:

$$G^k(\mathbf{u}^{k+1},\mathbf{v}) = G^k(\mathbf{u}^k,\mathbf{v}) - \gamma^k \left[A(\mathbf{u}^k,\mathbf{v}) + J'(\mathbf{u}^k,\mathbf{v}) - L(\mathbf{v}) \right], \ k = 0,1,\dots. \tag{18}$$

Note, that in general case iterative method (18) does not lead to domain decomposition. Let us propose such variants of this method, which involve the domain decomposition. At first, let us take bilinear forms G^k in method (18) as follows:

$$G^k(\mathbf{u},\mathbf{v}) = \partial^2 F_1(\mathbf{u}^k,\mathbf{u},\mathbf{v}) = A(\mathbf{u},\mathbf{v}) + \partial^2 J(\mathbf{u}^k,\mathbf{u},\mathbf{v}), \ \mathbf{u},\mathbf{v} \in V_0, \tag{19}$$

$$\partial^2 J(\mathbf{u}^k,\mathbf{u},\mathbf{v}) = \sum_{\{\alpha,\beta\} \in Q} \int_{S_{\alpha\beta}} \chi_{\alpha\beta}^k \, g'_{\alpha\beta}(d_{\alpha\beta} - u_{\alpha n}^k - u_{\beta n}^k) \left[u_{\alpha n} + u_{\beta n} \right] \left[v_{\alpha n} + v_{\beta n} \right] dS,$$

$$\chi_{\alpha\beta}^k = -[\operatorname{sgn}(d_{\alpha\beta} - u_{\alpha n}^k - u_{\beta n}^k)]^- = \{ 0, d_{\alpha\beta} - u_{\alpha n}^k - u_{\beta n}^k \ge 0 \} \vee \{ 1, \text{else} \}. \tag{20}$$

Here $\partial^2 F_1(\mathbf{u}^k,\mathbf{u},\mathbf{v})$, $\partial^2 J(\mathbf{u}^k,\mathbf{u},\mathbf{v})$ are one of the second subdifferentials of functionals F_1 and J in point $\mathbf{u}^k \in V_0$. In the case when $\gamma^k = 1$, $k = 0,1,\ldots$, iterative method (18) with bilinear forms (19) corresponds to semismooth Newton method for variational equation (15). However, this method does not lead to domain decomposition.

Now, let us take bilinear forms G^k in the following way:

$$G^k(\mathbf{u},\mathbf{v}) = A(\mathbf{u},\mathbf{v}) + X^k(\mathbf{u},\mathbf{v}),\ \ \mathbf{u},\mathbf{v} \in V_0, \tag{21}$$

$$X^k(\mathbf{u},\mathbf{v}) = \sum_{\alpha=1}^{N} \sum_{\beta \in B_\alpha} \int_{S_{\alpha\beta}} \psi^k_{\alpha\beta}\, g'_{\alpha\beta}(d_{\alpha\beta} - u^k_{\alpha n} - u^k_{\beta n})\, u_{\alpha n} v_{\alpha n}\, dS,\ \ \mathbf{u},\mathbf{v} \in V_0, \tag{22}$$

where $\psi^k_{\alpha\beta}(\mathbf{x}) = \{\, 1, \mathbf{x} \in S^k_{\alpha\beta} \,\} \vee \{\, 0, \mathbf{x} \in S_{\alpha\beta} \backslash S^k_{\alpha\beta} \,\}$ are characteristic functions of some given subsets $S^k_{\alpha\beta} \subseteq S_{\alpha\beta}$ of possible contact areas.

Iterative method (18) with bilinear forms (21) can be written in such way:

$$A(\tilde{\mathbf{u}}^{k+1},\mathbf{v}) + X^k(\tilde{\mathbf{u}}^{k+1},\mathbf{v}) = L(\mathbf{v}) + X^k(\mathbf{u}^k,\mathbf{v}) - J'(\mathbf{u}^k,\mathbf{v}),\ \ \forall \mathbf{v} \in V_0. \tag{23}$$

$$\mathbf{u}^{k+1} = \gamma^k \tilde{\mathbf{u}}^{k+1} + (1-\gamma^k)\,\mathbf{u}^k,\ \ k = 0,1,\ldots. \tag{24}$$

Since the common quantities of the subdomains are known from the previous iteration, variational equation (23) splits into N separate equations in subdomains Ω_α, and iterative method (23)–(24) can be written in the following equivalent form:

$$\begin{aligned} a_\alpha(\tilde{\mathbf{u}}^{k+1}_\alpha,\mathbf{v}_\alpha) + \sum_{\beta \in B_\alpha} \int_{S_{\alpha\beta}} \psi^k_{\alpha\beta}\, g'_{\alpha\beta}(d_{\alpha\beta} - u^k_{\alpha n} - u^k_{\beta n})\, \tilde{u}^{k+1}_{\alpha n} v_{\alpha n}\, dS = \\ = l_\alpha(\mathbf{v}_\alpha) + \sum_{\beta \in B_\alpha} \int_{S_{\alpha\beta}} \psi^k_{\alpha\beta}\, g'_{\alpha\beta}(d_{\alpha\beta} - u^k_{\alpha n} - u^k_{\beta n})\, u^k_{\alpha n} v_{\alpha n}\, dS + \\ + \sum_{\beta \in B_\alpha} \int_{S_{\alpha\beta}} g^-_{\alpha\beta}(d_{\alpha\beta} - u^k_{\alpha n} - u^k_{\beta n})\, v_{\alpha n}\, dS,\ \ \forall \mathbf{v}_\alpha \in V^0_\alpha, \end{aligned} \tag{25}$$

$$\mathbf{u}^{k+1}_\alpha = \gamma^k \tilde{\mathbf{u}}^{k+1}_\alpha + (1-\gamma^k)\,\mathbf{u}^k_\alpha,\ \ \alpha = 1,2,\ldots,N,\ \ k = 0,1,\ldots. \tag{26}$$

In each iteration k of method (25)–(26), we have to solve N linear variational equations (25) in parallel, which correspond to linear elasticity problems in separate bodies Ω_α with Robin boundary conditions on possible contact areas. Therefore, this method refers to parallel Robin–Robin type domain decomposition schemes.

By taking different characteristic functions $\psi^k_{\alpha\beta}$, we can obtain different particular cases of domain decomposition method (25)–(26). Thus, taking $\psi^k_{\alpha\beta}(\mathbf{x}) \equiv 0$ ($S^k_{\alpha\beta} = \emptyset$), $\forall \alpha,\beta$, $\forall k$, we get parallel Neumann–Neumann domain decomposition scheme. Other borderline case is when $\psi^k_{\alpha\beta}(\mathbf{x}) \equiv 1$ ($S^k_{\alpha\beta} = S_{\alpha\beta}$), $\forall \alpha,\beta$, $\forall k$.

Moreover, we can choose characteristic functions $\psi^k_{\alpha\beta}$ by formula (20), i.e. $\psi^k_{\alpha\beta} = \chi^k_{\alpha\beta}$. Numerical experiments, provided by us, have shown, that such DDM has higher convergence rate than other particular domain decomposition schemes.

6 Numerical analysis

Numerical analysis of proposed DDMs has been provided for plane problem of unilateral contact between two isotropic bodies Ω_1 and Ω_2, one of which has a groove (Fig.1b). The bodies are loaded by normal stress with intensity $q = 10\,\text{MPa}$. Each body has length $l = 4\,\text{cm}$ and height $h = 1\,\text{cm}$. The elasticity constants of the bodies are the same: $E_1 = E_2 = 2.1 \cdot 10^5\,\text{MPa}$, $\nu_1 = \nu_2 = 0.3$. The distance between bodies is $d_{12}(\mathbf{x}) = r\left\{[1-(x_1-l)^2/b^2]^+\right\}^{3/2}$, $\mathbf{x} \in S_{12}$, where $b = 1\,\text{cm}$, $r = 5 \cdot 10^{-4}\,\text{cm}$.

Across possible contact area S_{12} there is a nonlinear Winkler layer. The relationship between normal contact stresses and displacements of this layer is described by the following power function: $g_{12}(w_{12}(\mathbf{x})) = B^{-1/a}\,\text{sgn}(w_{12}(\mathbf{x}))\,|w_{12}(\mathbf{x})|^{1/a}$, $\mathbf{x} \in S_{12}$, where parameters B and a are taken from the intervals $B \in [10^{-6}\,\text{cm}/(\text{MPa})^a, 2 \cdot 10^{-4}\,\text{cm}/(\text{MPa})^a]$, $a \in [0.1, 1]$. For such choice of these parameters the nonlinear Winkler layer models a roughness of the possible contact surface [6].

This problem has been solved by DDM (25)–(26) with stationary iterative parameters $\gamma^k = \gamma$, $\forall k$ and characteristic functions ψ_{12}^k, taken by formula (20), i.e. $\psi_{12}^k = \chi_{12}^k$, $\forall k$. For solving linear variational problems (25) in each iteration k we have used finite element method with 8192 linear triangular elements for each body.

We have used the following initial guesses for displacements $u_{1n}^0(\mathbf{x}) = u_{2n}^0(\mathbf{x}) \equiv 10^{-4}\,\text{cm}$, and the next stopping criterion: $\rho_\alpha^{k+1} = \left\|u_{\alpha n}^{k+1} - u_{\alpha n}^k\right\|_2 / \left\|u_{\alpha n}^{k+1}\right\|_2 \le \varepsilon_u$, $\alpha = 1,2$, where $\|u_{\alpha n}\|_2 = \sqrt{\sum_j [u_{\alpha n}(\mathbf{x}^j)]^2}$ is discrete norm, $\mathbf{x}^j \in S_{12}$ are finite element nodes on the possible contact area, and $\varepsilon_u > 0$ is relative accuracy.

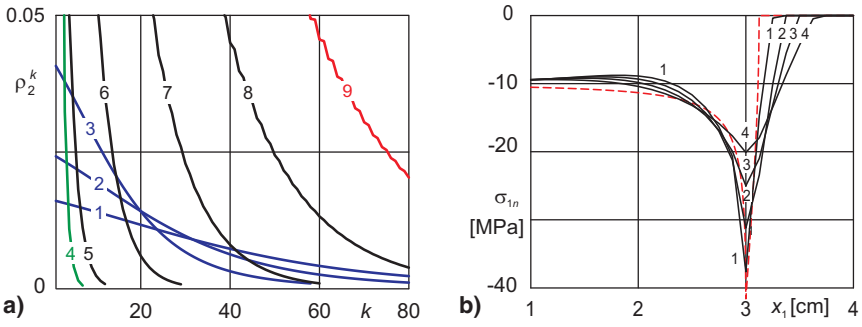


Fig. 2 Relative error (a), and normal contact stress (b)

At Fig.2a the relative error ρ_2^k of displacement u_{2n} on different iterations k, obtained for $B = 2.5 \cdot 10^{-5}\,\text{cm}/(\text{MPa})^a$, $a = 0.5$, is represented for different values of parameter γ. Curves 1–9 correspond to $\gamma = 0.02, 0.03, 0.05, 0.6, 0.8\,(0.3), 0.9, 0.95, 0.97, 0.98$. For these values of parameter γ, DDM (25)–(26) reaches the accuracy $\varepsilon_u = 10^{-3}$ in 110, 83, 58, 7, 12 (14), 29, 60, 102, 155 iterations respectively. Thus, we conclude, that the best convergence rate reaches if $\gamma = 0.6$. The convergence rate is good if $\gamma \in [0.1, 0.9]$. However, it becomes slow when γ is close to 0 or to 1. For $\gamma = 0.98$ the method is still convergent, but the convergence becomes nonmonotone.

We also have established, that the convergence rate of proposed DDMs does not deppend strongly on the number of finite element nodes m in each body. For $m = 43$, 149, 553, 2129, 8353, and 33089, DDM (25)–(26) with parameter $\gamma = 0.6$ reaches the accuracy $\varepsilon_u = 10^{-6}$ in 15, 15, 14, 14, 14, and 14 iterations respectively.

At Fig.2b the normal contact stress $\sigma_{1n} = \sigma_{2n}$, obtained by DDM (25)–(26) for $B = 10^{-5}\,\mathrm{cm}/(\mathrm{MPa})^a$ and different values of parameters a is represented. Curves 1–4 correspond to numerical solution for $a = 0.3$, 0.6, 0.8, 1. Dashed curve represents the analytical solution, obtained in [13] for contact between two halfspaces without nonlinear layer. Here we conclude, that for small values of a ($a \leq 0.3$) the influence of nonlinear layer on the contact behavior is not so large and the numerical solutions are close to the solution without layer. However, for larger values of a ($a \geq 0.5$) the influence of nonlinear layer becomes more significant and can not be neglected.

Acknowledgements This work was partially supported by Grant 23-08-12 of National Academy of Sciences of Ukraine

References

1. Bayada, G., Sabil, J., Sassi, T.: A Neumann–Neumann domain decomposition algorithm for the Signorini problem. Applied Mathematics Letters **17**(10), 1153–1159 (2004)
2. Bresch, D., Koko, J.: An optimization-based domain decomposition method for nonlinear wall laws in coupled systems. Math. Models Methods Appl. Sci. **14**(7), 1085–1101 (2004)
3. Dostál, Z., Kozubek, T., Vondrák, V., Brzobohatý, T., Markopoulos, A.: Scalable TFETI algorithm for the solution of multibody contact problems of elasticity. Int. J. Numer. Methods Engrg. **41**, 675–696 (2010)
4. Dyyak, I.I., Prokopyshyn, I.I.: Domain decomposition schemes for frictionless multibody contact problems of elasticity. In: G.K. et al. (ed.) Numerical Mathematics and Advanced Applications 2009, pp. 297–305. Springer (2010)
5. Dyyak, I.I., Prokopyshyn, I.I., Prokopyshyn, I.A.: Penalty Robin–Robin domain decomposition methods for unilateral multibody contact problems of elasticity: Convergence results (2012). URL `http://arxiv.org/pdf/1208.6478.pdf`
6. Goryacheva, I.G.: Contact mechanics in tribology. Kluwer (1998)
7. Hintermüller, M., Ito, K., Kunisch, K.: The primal-dual active set strategy as semismooth Newton method. SIAM J. OPTIM. **13**(3), 865–888 (2003)
8. Koko, J.: Convergence analysis of optimization-based domain decomposition methods for a bonded structure. Applied Numerical Mathematics (58), 69–87 (2008)
9. Koko, J.: Uzawa block relaxation domain decomposition method for a two-body frictionless contact problem. Applied Mathematics Letters **22**, 1534–1538 (2009)
10. Prokopyshyn, I.I.: Parallel domain decomposition schemes for frictionless contact problems of elasticity. Visnyk Lviv Univ. Ser. Appl. Math. Comp. Sci. **14**, 123–133 (2008). [In Ukrainian]
11. Prokopyshyn, I.I., Dyyak, I.I., Martynyak, R.M., Prokopyshyn, I.A.: Penalty Robin–Robin domain decomposition schemes for contact problems of nonlinear elasticity. Lect. Notes Comput. Sci. Eng. **91**, 647–654 (2013). [Accepted to DD20 Proceedings]
12. Sassi, T., Ipopa, M., Roux, F.X.: Generalization of Lion's nonoverlapping domain decomposition method for contact problems. Lect. Notes Comput. Sci. Eng. **60**, 623–630 (2008)
13. Shvets, R.M., Martynyak, R.M., Kryshtafovych, A.A.: Discontinuous contact of an anisotropic half-plane and a rigid base with disturbed surface. Int. J. Engng. Sci. **34**(2), 183–200 (1996)
14. Suquet, P.M.: Discontinuities and plasticity. In: CISM Courses Lect., 302, pp. 279–340 (1988)

15. Vorovich, I.I., Alexandrov, V.M. (eds.): Contact Mechanics. Fizmatlit, Moscow (2001)
16. Wohlmuth, B.: Variationally consistent discretization schemes and numerical algorithms for contact problems. Acta Numerica **20**, 569–734 (2011)

Asymptotic expansions and domain decomposition

G. Geymonat[1], S. Hendili[2,3], F. Krasucki[3], M. Serpilli[4] and M. Vidrascu[2]

1 Introduction

At a first glance asymptotic expansions and domain decomposition are two alternatives to efficiently solve multi scale elasticity problems. In this paper we will combine these two methods: we will use, for several types of problems, asymptotic expansions and show that for an efficient implementation of problems obtained at the asymptotic limit it may be useful to use domain decomposition type algorithms. In particular we will consider problems with heterogenous or non heterogenous thin layers(see Fig 1 a) et b)). To directly solve such problems by a standard finite element method is too expensive from a computational point of view. That is why specific asymptotic expansions are used and allow to replace the original problem by a set of problems defined on a new domain where the thin layer is replaced by a line in 2D or a surface in 3D (see Fig 1 c)). In addition particular jumping conditions are defined on this new interface yielding a non standard problem which can be solved by a Neumann-Neumann domain decomposition algorithm. The paper is organized as follows: In Section 2 we review of a domain decomposition algorithm on an elasticity problem, in Section 3 we consider a thin layer of heterogeneities which can be holes or elastic inclusions and, finally, in Section 4 we consider a multi-materials with a thin layer with high ratio in material properties.

2 Domain decomposition algorithm: general setting for an elasticity problem

The aim of this paragraph is to specify the notations. We consider a standard linear elasticity problem:

$$\begin{cases} div\sigma^{\varepsilon} = 0 & \text{in } \Omega^{\varepsilon} \\ \sigma^{\varepsilon} = \mathsf{A}e(u^{\varepsilon}) & \text{in } \Omega^{\varepsilon} \\ \sigma^{\varepsilon} n = F & \text{on } \Gamma_F \\ u^{\varepsilon} = 0 & \text{on } \Gamma_0 \end{cases} \tag{1}$$

[1] LMS, UMR-CNRS 7649, Ecole Polytechnique e-mail: giuseppe.geymonat@lms.polytechnique.fr .[2] EPI REO, INRIA Rocquencourt, e-mail: {soufiane.hendili}{marina.vidrascu}@inria.fr, .[3] I3M, UMR-CNRS 5149, Université Montpellier 2 e-mail: krasucki@math.univ-montp2.fr .[4] Departement of Civil and Building construction engineering, and architecture, Università Politecnica delle Marche, Ancona e-mail: m.serpilli@univpm.it

The mechanical characteristics of the multi-material structure are described by the elasticity tensor A. Each material is isotropic but A is indeed material dependent. In the sequel we will omit this constitutive equation. The structure is clamped on a part $\Gamma_0 \subset \partial\Omega$ (of surface measure > 0) and a density F of surface forces is applied on the complementary part Γ_F. In a variational form this problem writes

$$A(\mathbf{u},v) = L(v) \text{ for all } v \in V, \text{ with } A(\mathbf{u},v) = \int_\Omega A^{ijk\ell} e_{k\ell}(\mathbf{u}) e_{ij}(v)\, dx. \qquad (2)$$

Let us mention that the variational form is always used to discretize the problem, nevertheless in order to simplify notations we will use either partial differential equations or variational form. The same problem will be considered in sections 3 and 4, where the the domain differs with respect to the heterogeneities. We will explain how the domain decomposition algorithm is adapted in each situation. In order to use a primal domain decomposition to solve the problem we transform the problem on the entire domain in a problem on the interface. After splitting the domain in non overlapping subdomains we introduce an additional unknown, $\lambda = Tr(u)$ on the interface. For simplicity reasons we will consider here only two sub-domains and only a first level preconditioner. To solve the original problem is equivalent to solving the following problem on each subdomain:

$$\begin{cases} div\sigma(u^i) = f^\Omega & \text{in } \Omega_i \\ \sigma n = f^\Gamma & \text{on } \partial\Omega_F \cap \Omega_i \\ u^i = u^d & \text{on } \partial\Omega_u \cap \Omega_i \\ u^i = \gamma & \text{on } \Gamma \end{cases} \qquad (3)$$

By linearity $u^i = u_0^i + u_\gamma^i$ where u_0^i is the solution of (3) with $u_0^i = 0$ on Γ and u_γ^i is the solution of (3) with $f^\Omega = 0, f^\Gamma = 0$. In order to settle the interface problem we write the continuity of the normal stress on the interface:

$$\sigma(u^1)n^1 + \sigma(u^2)n^2 = \sigma(u_\gamma^1)n^1 + \sigma(u_0^1)n^1 + \sigma(u_\gamma^2)n^2 + \sigma(u_0^2)n^2 = 0$$

Using the Steklov Poincaré operator S which is defined as follows: for γ given on Γ (the sub-domains interface)

$$S_i\gamma = \sigma(u_\gamma^i)n^i$$

where n^i denotes the outer normal on Γ, the interface problem writes:

$$S_1\gamma + S_2\gamma = -\sigma(u_0^1)n^1 - \sigma(u_0^2)n^2 \qquad (4)$$

In variational form

$$S_1(\gamma,v) + S_2(\gamma,v) = -L(\sigma(u_0^1)n^1, v) - L(\sigma(u_0^2)n^2, v)$$

This problem will be solved using a iterative method, the preconditioner is $M = \alpha_1 S_1^{-1} + \alpha_2 S_2^{-1}$ with $\alpha_1 + \alpha_2 = 1$. ([6], [3])

The parallel between this approach and the one used in the asymptotic analysis (as described in 1) is that a particular problem has to be solved on the interface, the next sections will specify this concept.

3 Structure with a thin layer of heterogeneities

Let us consider a three-dimensional structure with small identical heterogeneities periodically distributed along a surface ω. Let ε be a small dimensionless parameter which characterizes the diameter and the periodic arrangement of the heterogeneities. We denote B^ε the layer of thickness ε containing the heterogeneities centered on ω (see Fig. 1 a)).

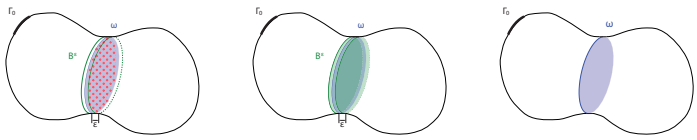


Fig. 1 a) Heterogeneous layer b) Homogeneous layer c) Limit domain

The domain Ω contains I^ε the set of identical heterogeneities of diameter εD and ε-periodically distributed in the vicinity of the interior surface ω of equation $x_1 = 0$. We consider the problem (3) with two types of inclusions: cavities and elastic inclusions. The displacement field u^ε and the stress field σ^ε, satisfy, respectively, equilibrium equation (1).

Notice that Ω is a domain with a number of heterogeneities which depends on ε. For the elastic inclusions A^S and A^I (the elasticity tensor in the structure, respectively in the inclusions) are of same order of magnitude.

The asymptotic analysis of this problem for $\varepsilon \to 0$ provides a model describing the linear elastic behavior of the structure on a simplified domain denoted by Ω_0 where the layer B^ε becomes the surface Γ (see Fig. 1 c)). More precisely, by assuming that $u^\varepsilon \simeq u^0 + \varepsilon u^1$, the initial problem (1) is approximated by two new ones where the layer of heterogeneities is replaced by a surface on which particular jump conditions are defined.

The zeroth order approximation u^0 is the solution of the following transmission linear problem :

$$\begin{cases} div\sigma^0 = 0 & \text{in } \Omega_0 \\ \sigma^0 n = F & \text{on } \Gamma_F \\ u^0 = 0 & \text{on } \Gamma \end{cases} \tag{5}$$

Notice that there are no jumps on Γ for the outer approximation. In other words, at the zero order the outer approximation does not consider the heterogeneities. Thus this problem can be solved using a standard finite element procedure.

The first order approximation u^1 is the unique solution of the following boundary value problem (with transmissions conditions on Γ):

$$\begin{cases} div\sigma^1 & = 0 & \text{in } \Omega_0 \backslash \Gamma \\ \sigma^1 n & = 0 & \text{on } \Gamma_F \\ u^1 & = 0 & \text{on } \Gamma_0 \\ \left[u^1\right](\hat{x}) & = \mathscr{G}_d\left(u^0(0,\hat{x})\,;\left[V^{ij}\right]^{\infty}\right) \\ \left[\sigma^1 e_1\right](\hat{x}) & = \mathscr{G}_{nS}\left(u^0(0,\hat{x})\,;\int_Y T^{ij}(y)\,dy\right) \end{cases} \tag{6}$$

where V^{ij} are the solutions of nine elementary problems defined on one representative cell Y ([5],[4]) and T^{ij} are the stress fields associated with V^{ij} and $\left[V^{ij}\right]^{\infty} = \lim_{y_1\to+\infty} V^{ij} - \lim_{y_1\to-\infty} V^{ij-}$

$\mathscr{G}_d$ has the same structure for the different types of inclusions, while $\mathscr{G}_{nS}$ depends on the inclusion:

$$\mathscr{G}_d = \frac{\partial u_i^0}{\partial x_j}(0,\hat{x})\left[V^{ij}\right]^{\infty} \tag{7}$$

i) in the elastic inclusions case one has:

$$\mathscr{G}_{nS} = div\left(|I|\left(\mathsf{A}^S - \mathsf{A}^I\right)\mathsf{e}\left(u^0(0,\hat{x})\right) - \frac{\partial u_i^0}{\partial x_j}(0,\hat{x})\int_Y T^{ij}(y)\,dy\right) \tag{8}$$

ii) in the cavities case one has:

$$\mathscr{G}_{nS} = div\left(|I|\mathsf{A}^I\mathsf{e}\left(u^0(0,\hat{x})\right) - \frac{\partial u_i^0}{\partial x_j}(0,\hat{x})\int_Y T^{ij}(y)\,dy\right) \tag{9}$$

Let us emphasize that, for the first order problem, $\mathscr{G}_d$ and $\mathscr{G}_{nS}$ are given and depend on the first and second order derivatives of the zeroth order problem. This is not an issue at the domain decomposition level, while, at the implementation level, since the solution u^0 is *only* of class C^0, a regularization is needed. In practice, an efficient way to implement the jump conditions in problem (6) is to solve this problem by a domain decomposition type algorithm which will be detailed hereafter.

Finally, the generic form of the first order problem, (6) is given by:

$$\begin{cases} -div\sigma(u) = 0 & \text{in } \Omega \\ \sigma n = 0 & \text{on } \partial\Omega_F \\ u = 0 & \text{on } \partial\Omega_u \\ [u] = \mathscr{G}_d & \text{on } \Gamma \\ [\sigma n] = \mathscr{G}_{nS} & \text{on } \Gamma \end{cases} \tag{10}$$

where $\mathscr{G}_d$ and $\mathscr{G}_{nS}$ denote, respectively, the gap in displacements and normal stresses on Γ. By using the linearity of the problem, we will search, in each subdomain a solution of the form

$$u^i = w^i + \beta_i z^i$$

where β_i are two real numbers conveniently chosen and z^i are the solutions of the following two independent problems:

$$\begin{cases} -div\sigma(z^i) = 0 & \text{in } \Omega_i \\ \sigma n = 0 & \text{on } \partial\Omega_F \cap \Omega_i \\ z^i = 0 & \text{on } \partial\Omega_u \cap \Omega_i \\ z^i = \mathscr{G}_d & \text{on } \Gamma \end{cases} \tag{11}$$

Notice that

$$-div\sigma(w^i) = -div\sigma(u^i - \beta_i z^i) = 0$$

The transmission conditions for w^i are given by:

$$\begin{cases} [w] = [u] - \beta_1\mathscr{G}_d + \beta_2\mathscr{G}_d = (1 - \beta_1 + \beta_2)\mathscr{G}_d \\ [\sigma n] = [u] + \beta_1\sigma(z^1)n - \beta_2\sigma(z^2)n = \mathscr{G}_{nS} + \beta_1\sigma(z^1)n - \beta_2\sigma(z^2)n \end{cases}$$

If we choose $1 - \beta_1 + \beta_2 = 0$ then w is continuous on the interface Γ, while the normal stress is discontinuous at the interface.

By introducing the Steklov Poincaré, as described above, the unknown γ on the interface is the solution of the following problem:

$$(S_1 + S_2)\gamma = -\sigma(w_0^1)n^1 - \sigma(w_0^2)n^2 + \mathscr{G}_{nS} + \beta_1\sigma(z^1)n^1 - \beta_2\sigma(z^2)n^2$$

Let us remark that this equation differs from (4) only on the right hand side. In this situation the solution of the entire problem is not as regular as in section 2. Here, because of the jumps, the solution is not in $H^1(\Omega)$, this is why the norms used in the following numerical simulations are $L^2(\Omega)$. Thus as the operator does not change, the same algorithms (and in particular the same preconditioner) may be used to solve the problem with the same performance and no additional analysis is required to prove efficiency.

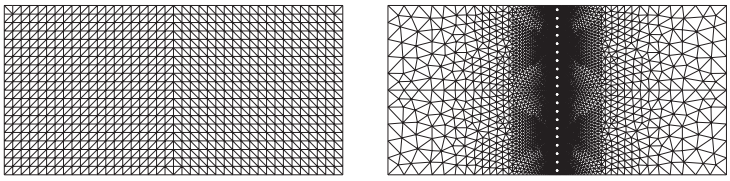


Fig. 2 a) Mesh used for the asymptotic computation b) Fine mesh for $\varepsilon = \frac{1}{20}$

In order to numerically validate this approach we consider a 2D case where Ω is a plane domain containing $\mathscr{N}^\varepsilon$ holes of diameter εD Notice that the domain and thus the number of holes depends on ε. A reference solution u_h^ε of the problem (1) is computed on a large mesh (see Fig 2 b)) and compared with the asymptotic solution u_h^0 and $u_h^0 + \varepsilon u_h^1$ obtained by solving the problems (5) and (6) on a corse mesh (see Fig. 2 a)). This comparison is performed by computing the relative error for the L^2-norm (see table (1)).

Table 1 L^2-errors norms computed in Ω^ε

ε	Nb elements	dofs	$\frac{\|\|u_h^\varepsilon - u_h^0\|\|_{L^2}}{\|\|u_h^\varepsilon\|\|_{L^2}}$	$\frac{\|\|u_h^\varepsilon - (u_h^0 + \varepsilon u_h^1)\|\|_{L^2}}{\|\|u_h^\varepsilon\|\|_{L^2}}$
1/20	13348	54938	0.013501216	0.001225971
1/40	27668	113530	0.006689361	0.000475813
1/80	57164	234050	0.003281498	0.000176916

4 Multimaterials with strong curved interface

In this section we analyze the mechanical behavior of a particular structural assembly, which is constituted by an elastic shell-like inclusion with high rigidity surrounded by two three-dimensional elastic bodies.

Let Ω^+ and Ω^- be two disjoint open domains with smooth boundaries $\partial\Omega^+$ and $\partial\Omega^-$. Let $\omega := \{\partial\Omega^+ \cap \partial\Omega^-\}^\circ$ be the interior of the common part of the boundaries which is assumed to be a non empty domain in $\mathbb{R}^2$. Let $\theta \in \mathscr{C}^2(\overline{\omega};\mathbb{R}^3)$ be an immersion such that the vectors $\mathbf{a}_\alpha(y) := \partial_\alpha\theta(y)$ form the covariant basis of the tangent plane to the surface $S := \theta(\overline{\omega})$. We note with $\mathbf{a}_3(y) := \frac{\mathbf{a}_1(y)\wedge\mathbf{a}_2(y)}{|\mathbf{a}_1(y)\wedge\mathbf{a}_2(y)|}$ the unit normal vector to S. We insert an intermediate curved layer moving Ω^+ and Ω^- in the $\mathbf{a}_3$ and $-\mathbf{a}_3$ directions, respectively, by an amount equal to $t^\varepsilon > 0$, where ε is a small dimensionless real parameter. Then let $\Omega^{\pm,\varepsilon} := \{x^\varepsilon := x \pm t^\varepsilon\mathbf{a}_3;\ x \in \Omega^\pm\}$, $\Omega^{m,\varepsilon} := \omega\times]-t^\varepsilon, t^\varepsilon[$, and $\Omega^\varepsilon := \Omega^{-,\varepsilon} \cup \Omega^{+,\varepsilon} \cup \Omega^{m,\varepsilon}$, as shown in Fig. 1 The structure is clamped on $\Gamma_0^\varepsilon \subset (\partial\Omega^\varepsilon \setminus \Gamma^{m,\varepsilon})$. We consider that S coincides with the middle surface of the shell-like inclusion $\Omega^{m,\varepsilon}$. Moreover, the shell thickness t^ε depends linearly on ε, so that $t^\varepsilon = \varepsilon t$. For a more detailed treatment of this asymptotic problem in a general curvilinear framework, the reader can refer to [1], [2].

The physical variational problem defined over the variable domain Ω^ε is

$$\begin{cases} \text{Find } \mathbf{u}^\varepsilon \in V^\varepsilon := \{v^\varepsilon \in H^1(\Omega^\varepsilon;\mathbb{R}^3);\ v^\varepsilon|_{\Gamma_0^\varepsilon} = \mathbf{0}\} \\ A_-^\varepsilon(\mathbf{u}^\varepsilon, v^\varepsilon) + A_+^\varepsilon(\mathbf{u}^\varepsilon, v^\varepsilon) + A_m^\varepsilon(\mathbf{u}^\varepsilon, v^\varepsilon) = L(v^\varepsilon) \text{ for all } v^\varepsilon \in V^\varepsilon, \end{cases} \tag{12}$$

where A is defined as in (2).

The functional $L(\cdot)$ is the linear form associated with the applied forces. Here $A^{ijk\ell,\varepsilon} := \lambda^\varepsilon g^{ij,\varepsilon} g^{k\ell,\varepsilon} + \mu^\varepsilon(g^{ik,\varepsilon} g^{j\ell,\varepsilon} + g^{i\ell,\varepsilon} g^{jk,\varepsilon})$ are the contravariant components of the elasticity tensor, where g^{ij} can be considered as the curvilinear version of the Kronecker's delta. Let us suppose that the Lamé's constants of the isotropic materials satisfy the following dependences with respect to ε: $\lambda^{\pm,\varepsilon} = \lambda^\pm$, $\mu^{\pm,\varepsilon} = \mu^\pm$, $\lambda^{m,\varepsilon} = \frac{1}{\varepsilon}\lambda^m$, $\mu^{m,\varepsilon} = \frac{1}{\varepsilon}\mu^m$.

As shown in [1], the asymptotic expansion method applied to the physical problem (12) leads to a simplified model for the assembly, in which the layer inclusion is reduced to its middle surface as ε tends to zero. Thus the presence of the layer is replaced by a surface shell like energy at the interface which corresponds to a partic-

ular membrane transmission condition between the two three-dimensional bodies. The main result is contained in the following theorem:

Theorem 1. *The leading term* $\mathbf{u}^0$ *of the asymptotic expansion* $\mathbf{u}(\varepsilon) = \mathbf{u}^0 + \varepsilon\mathbf{u}^1 + \varepsilon^2\mathbf{u}^2 + ...$, *is the unique solution of the following limit problem:*

$$\begin{cases} \textit{Find } \mathbf{u}^0 \in V_M \textit{ such that} \\ A^-(\mathbf{u}^0, v) + A^+(\mathbf{u}^0, v) + A_M^m(\mathbf{u}^0, v) = L(v) \;\; \textit{for all } v \in V_M \end{cases} \tag{13}$$

where $V_M := \{v \in H^1(\Omega^+ \cup \omega \cup \Omega^-; \mathbb{R}^3);\ v|_\omega \in H^1(\omega; \mathbb{R}^2) \times H^{\frac{1}{2}}(\omega),\ v|_{\Gamma_0} = \mathbf{0}\}$, *and*

$$A_M^m(\mathbf{u}^0, v) = 2t \int_\omega a^{\alpha\beta\sigma\tau} e_{\sigma\tau}(\mathbf{u}^0) e_{\alpha\beta}(v)\, dy, \tag{14}$$

is the bilinear form associated with the membrane behavior of the shell, $a^{\alpha\beta\sigma\tau}$ *is the elasticity tensor of the shell and* $e_{\alpha\beta}(\mathbf{u}) := \frac{1}{2}(u_{\beta|\alpha} + u_{\alpha|\beta})$ *is the change of metric tensor.*

Remark. In the simplified model we obtain a membrane transmission condition at the interface between the two three-dimensional bodies, which can be interpreted as a curvilinear generalization of the Ventcel-type transmission condition obtained in [1]. Indeed, by integrating by parts problem (13), one has

$$\begin{cases} -\text{div } \sigma_\pm = \mathbf{f} \text{ in } \Omega^\pm, \\ \mathbf{u}^0 = \mathbf{0} \qquad \text{on } \Gamma_0, \end{cases} \qquad \begin{cases} \left[\sigma^{\alpha 3}\right] = \text{div } (n^{\alpha\beta}) \text{ in } \omega, \\ \left[\sigma^{33}\right] = n^{\alpha\beta} b_{\alpha\beta} \quad \text{in } \omega, \end{cases} \tag{15}$$

where $\sigma_\pm^{ij} := A_\pm^{ijk\ell} e_{k\ell}(\mathbf{u}^0)$ and $n^{\alpha\beta} := 2ta^{\alpha\beta\sigma\tau} e_{\sigma\tau}(\mathbf{u}^0|_\omega)$ represent, respectively, the Cauchy stress tensor and the membrane stress tensor of the shell, $\left[\sigma^{i3}\right] := \sigma_+^{i3} - \sigma_-^{i3}$ represents the stress jump at the interface ω, and $b_{\alpha\beta}$ is the second fundamental form associated to the shell middle surface.

In order to solve the problem (13) we introduce a specific domain decomposition algorithm, more precisely, we construct the interface problem. We consider three subdomains $\Omega^+ := \Omega^{(1)}$, $\Omega^- := \Omega^{(2)}$, and the shell Ω^m. For the two 3D domains, Ω^1, Ω^2 we introduce the corresponding Steklov Poincaré operator and we observe that the domain Ω^3 is the interface. Thus, in a variational form, the compatibility condition on the interface writes :

$$S^1(\gamma, v) + S^2(\gamma, v) + A_M^m(\gamma, v) = L(-\sigma(u_0^1)n^1 - L\sigma(u_0^2)n^2, v) \tag{16}$$

This problem can be solved by a Neumann-Neumann algorithm as well because, compared to (4) we add in the right hand side a term wich is symmetric and positive defined.

As a numerical example, we consider an axisymmetric problem of two thick cylinders bonded together with a cylindrical shell with high rigidity subjected to an internal pressure ($E_{cyl} = 5e05$, $E_{shell} = 5e07$, $\nu = 0.3$, t=0.1, $R_{max} = 6$). We choose this particular geometry because it is characterized by an immediate mechanical interpretation. Moreover we can compute an exact solution for this problem. We tested the domain decomposition by using two subdomains (the shell is "glued"

to another subdomain) and three subdomains and by studying the influence of a Neumann-Neumann preconditioner on the number of iterations. The preliminary results are shown in the following table. As we can see the number of iterations decreases drastically when adopting a preconditioner.

Table 2 Mesh: $N_{el} = 11020$, $N_{el,shell} = 580$

Subdomains	Iterations	Iterations with preconditioner
2	69	6
2+1(shell)	70	46

In the actual simulations we can use membrane or shell elements. The shell is more robust but also more computationally demanding. In our example we used a membrane element. The drawback is that the operator is not invertible (that is needed in the preconditioning step) and that explains why the results with two domains are far better than with three domains. Hence, our test example does not behave totally as a pure membrane. This feature disappears when shell elements are used or when the problem has a pure membrane behavior.

Acknowledgements This work was partially supported by the French Agence Nationale de la Recherche (ANR) under Grant Epsilon (BLAN08-2 312370) (Domain decomposition and multi-scale computations of singularities in mechanical structures).

References

1. Bessoud, A.L., Krasucki, F., Serpilli, M.: Asymptotic analysis of shell-like inclusions with high rigidity. J. Elasticity **103**(2), 153–172 (2011). DOI 10.1007/s10659-010-9278-1. URL http://dx.doi.org/10.1007/s10659-010-9278-1
2. Chapelle, D., Ferent, A.: Modeling of the inclusion of a reinforcing sheet within a 3D medium. Math. Models Methods Appl. Sci. **13**(4), 573–595 (2003). DOI 10.1142/S0218202503002635. URL http://dx.doi.org/10.1142/S0218202503002635
3. De Roeck, Y.H., Le Tallec, P., Vidrascu, M.: A domain-decomposed solver for nonlinear elasticity. Comput. Methods Appl. Mech. Engrg. **99**(2-3), 187–207 (1992). DOI 10.1016/0045-7825(92)90040-Q. URL http://dx.doi.org/10.1016/0045-7825(92)90040-Q
4. Geymonat, G., Hendili, S., Krasucki, F., Vidrascu, M.: The matched asymptotic expansion for the computation of the effective behavior of an elastic structure with a thin layer of holes. International Journal for Multiscale Computational Engineering **9**(5), 529–542 (2011). DOI 10.1615/IntJMultCompEng.v9.i5. URL http://hal.inria.fr/inria-00540992/en
5. Geymonat, G., Hendili, S., Krasucki, F., Vidrascu, M.: Matched asymptotic expansion method for an homogenized interface model. (2012). URL http://hal.archives-ouvertes.fr/hal-00757005. Submitted
6. Le Tallec, P.: Domain decomposition methods in computational mechanics. Comput. Mech. Adv. **1**(2), 121–220 (1994)

A Schur Complement Method for Compressible Two-Phase Flow Models

Thu-Huyen DAO[1,2], Michael NDJINGA[1], and Frédéric MAGOULÈS[2]

1 Introduction

Computations of complex two-phase flows are required for the safety analysis of nuclear reactors. These computations keep causing problems for the development of best estimate computer codes dedicated to design and safety studies of nuclear reactors. Moreover, we often need to find the long-term behavior of the system. In these cases, implicit schemes are proven very efficient. Unfortunately, for implicit schemes, after the discretization, we need to solve a nonlinear system $\mathscr{A}U = b$. This task is computationally expensive in particular since the matrix $\mathscr{A}$ is usually non-symmetric and very ill-conditioned. It is therefore necessary to find an efficient preconditioner.
When the size of the system is large, the parallel resolution on multiple processors is essential to obtain reasonable computation times. Currently in the thermal hydraulic code, FLICA-OVAP (see [7]), the matrix $\mathscr{A}$ and the right hand side b are stored on multiple processors and the system is solved in parallel with a Krylov solver with a classical incomplete factorization preconditioner. Unfortunately, the parallel preconditioners of FLICA-OVAP only perform well on a few processors. In contrast, if we want to increase the number of processors these parallel preconditioners perform poorly. Tests were run on different test cases and led us to conclude that it is often better not to use these parallel preconditioners, especially for $3D$ problems ([2]). This strategy does not make an optimal use of the available computational power. Hence, we seek for more efficient methods to distribute the computations. We study and use a domain decomposition method as an alternative to the classical distribution.

2 Mathematical model

For the modeling of two-phase flows, several sets of equations have been worked out. They range in complexity from the homogeneous equilibrium model to two-fluid models involving unequal pressure for each phase. In this paper, we consider the well-known two-fluid model. This model is obtained by averaging the balance equations for each separated phase, using space, time or ensemble averaged quantities (see [8] and [6]). The unknown physical quantities are the volume fraction

[1] CEA–Saclay, DEN, DM2S, STMF, LMEC, F–91191 Gif–sur–Yvette, France, [2] Appl. Mat. and Syst. Lab., Ecole Centrale Paris, 92295 Châtenay–Malabry, France, e-mail: frederic.magoules@hotmail.com

$\alpha_k \in [0,1]$, the density $\rho_k \geq 0$, and the velocity $\mathbf{u}_k$ of each phase. The subscript k stands for l if it is the liquid phase and g for the gas phase. The common averaged pressure of the two phases is denoted by p. In our model, pressure equilibrium between the two phases is postulated. For the sake of simplicity, we study the isentropic two-fluid model. This model can be written as follows:

$$\begin{cases} \frac{\partial(\alpha_g\rho_g)}{\partial t} + \nabla\cdot(\alpha_g\rho_g u_g) = 0, \\ \frac{\partial(\alpha_l\rho_l)}{\partial t} + \nabla\cdot(\alpha_l\rho_l u_l) = 0, \\ \frac{\partial(\alpha_g\rho_g u_g)}{\partial t} + \nabla\cdot(\alpha_g\rho_g u_g\otimes u_g) + \alpha_g\nabla p + \Delta p\nabla\alpha_g - \nabla\cdot(\alpha_g\nu_g\nabla u_g) = 0, \\ \frac{\partial(\alpha_l\rho_l u_l)}{\partial t} + \nabla\cdot(\alpha_l\rho_l u_l\otimes u_l) + \alpha_l\nabla p + \Delta p\nabla\alpha_l - \nabla\cdot(\alpha_l\nu_l\nabla u_l) = 0, \end{cases} \tag{1}$$

with $\alpha_g+\alpha_l=1$, and the two equations of state(EOS) $\rho_g=\rho_g(p)$ and $\rho_l=\rho_l(p)$. In our problem, we use the stiffened equation of state. Here ν_k is the viscosity of phase k, and Δp denotes the pressure default $p-p_k$ between the bulk average pressure and the interfacial average pressure.
By denoting $m_k=\alpha_k\rho_k$, $\mathbf{q}_k=\alpha_k\rho_k\mathbf{u}_k$ and $\mathbf{U}=(m_g,\mathbf{q}_g,m_l,\mathbf{q}_l)^t$, we can write the system (1) as follows:

$$\frac{\partial\mathbf{U}}{\partial t}+F^{conv}(\mathbf{U})+F^{diff}(\mathbf{U})=0, \quad \text{where} \tag{2}$$

$$F^{conv}(\mathbf{U})=\begin{pmatrix} \nabla\cdot\mathbf{q}_g \\ \nabla\cdot\mathbf{q}_l \\ \nabla\cdot(\mathbf{q}_g\otimes\frac{\mathbf{q}_g}{m_g})+\alpha_g\nabla p+\Delta p\nabla\alpha_g \\ \nabla\cdot(\mathbf{q}_l\otimes\frac{\mathbf{q}_l}{m_l})+\alpha_l\nabla p+\Delta p\nabla\alpha_l \end{pmatrix}, F^{diff}(\mathbf{U})=\begin{pmatrix} 0 \\ 0 \\ -\nabla\cdot(\alpha_g\nu_g\nabla\frac{\mathbf{q}_g}{m_g}) \\ -\nabla\cdot(\alpha_l\nu_l\nabla\frac{\mathbf{q}_l}{m_l}) \end{pmatrix}.$$

3 Numerical Method

Most of the numerical methods used in two-phase flow computer codes are based upon semi-implicit finite difference schemes with staggered grids and donor-cell differencing. The main features of these schemes are their efficiency and their robustness. However, these methods have a large amount of numerical dissipation, giving poor accuracy in smooth regions of the flow. Moreover, discontinutities are heavily smeared on coarse grids and oscillations appear when the grid is refined. Here, we propose to use an approximate Riemann solver to discretize and solve the system (2). We decompose the computational domain into N disjoint cells C_i with volume v_i. Two neighboring cells C_i and C_j have a common boundary ∂C_{ij} with area s_{ij}. We denote $N(i)$ the set of neighbors of a given cell C_i and n_{ij} the exterior unit normal vector of ∂C_{ij} . Integrating the system (2) over C_i and setting $\mathbf{U}_i(t)=\frac{1}{v_i}\int_{C_i}U(x,t)dx$ and $U_i^n=\mathbf{U}_i(n\Delta t)$, the discretized equations can be written:

$$\int_{C_i} \frac{\partial \mathbf{U}}{\partial t}\,\mathrm{d}\mathbf{x} + \sum_{j\in N(i)} \Phi_{ij}^{conv} + \sum_{j\in N(i)} \Phi_{ij}^{diff} = 0 \tag{3}$$

with Φ_{ij}^{conv}, Φ_{ij}^{diff} denote the numerical flux of convection and diffusion on the cell C_i in direction of the neighbor cell C_j.
The diffusion numerical flux Φ_{ij}^{diff} is approximated on structured meshes using the formula:

$$\Phi_{ij}^{diff} = D(\frac{\mathbf{U}_i + \mathbf{U}_j}{2})(\mathbf{U}_j - U_i). \tag{4}$$

Full details of the evaluation of diffusive flux terms are given in [16].
Due to the $\alpha_k \nabla p$ and $\Delta p \nabla \alpha_k$ terms, the inviscid part of the two-phase flow cannot be written in a conservative form. But this system can be written in the quasi-linear form:

$$\frac{\partial \mathbf{U}}{\partial t} + A(\mathbf{U})\frac{\partial \mathbf{U}}{\partial x} = 0. \tag{5}$$

Under some simplifying assumptions, the authors of [17] were able to obtain a conservative form that allowed them to give a sense to discontinous solutions. It was also under those assumptions that they have been able to develop an approximate Riemann solver of Roe-type for the system (5) providing a local linearization of the non-conservative term $\alpha_k \nabla p$. We can also contruct other linearizations than that of [17]. Here, we will not propose a specific linearization but a general method for the construction of the Roe matrix once we have chosen a linearization. We then define a local inviscid flux function $F^{loc}$and a local Roe matrix A_{Roe} for this linearization. The inviscid flux in the normal direction to the cell interface $\partial C_{i,j}$ is given by:

$$\begin{aligned} \Phi_{ij}^{conv} =& \frac{F^{loc}(\mathbf{U}_i) + F^{loc}(\mathbf{U}_j)}{2}.\mathbf{n}_{ij} + \mathscr{D}\frac{\mathbf{U}_i - \mathbf{U}_j}{2} \\ =& F^{loc}(\mathbf{U}_i).\mathbf{n}_{ij} + A^{-}(\mathbf{U}_j - \mathbf{U}_i), \end{aligned} \tag{6}$$

where $\mathscr{D}$ is an upwinding matrix, A_{Roe} the Roe matrix and $A^{\pm} = \frac{1}{2}(A_{Roe} \pm \mathscr{D})$.

The choice $\mathscr{D} = 0$ gives the centered scheme, whereas $\mathscr{D} = |A_{Roe}|$ gives the upwind scheme.

Newton scheme

Finally, since $\sum_{j\in N(i)} F^{loc}(\mathbf{U}_i).n_{ij} = 0$, using (6) and (4) the equation (3) of the numerical scheme becomes:

$$\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} + \sum_{j\in N(i)} \frac{s_{ij}}{v_i}\{(A^- + D)(\mathbf{U}_i^{n+1}, \mathbf{U}_j^{n+1})\}(\mathbf{U}_j^{n+1} - \mathbf{U}_i^{n+1}) = 0. \tag{7}$$

The system (7) is nonlinear, hence we use the following Newton iterative method to obtain the required solutions:

$$\frac{\delta \mathbf{U}_i^{k+1}}{\Delta t} + \sum_{j\in N(i)} \frac{s_{ij}}{v_i} \left[(A^- + D)(\mathbf{U}_i^k, \mathbf{U}_j^k)\right] \left(\delta \mathbf{U}_j^{k+1} - \delta \mathbf{U}_i^{k+1}\right)$$
$$= -\frac{\mathbf{U}_i^k - \mathbf{U}_i^n}{\Delta t} - \sum_{j\in N(i)} \frac{s_{ij}}{v_i} \left[(A^- + D)(\mathbf{U}_i^k, \mathbf{U}_j^k)\right] (\mathbf{U}_j^k - \mathbf{U}_i^k), \tag{8}$$

where $\delta \mathbf{U}_i^{k+1} = \mathbf{U}_i^{k+1} - \mathbf{U}_i^k$ is the variation of the k-th iterate that approximates the solution at time $n+1$. Defining the unknown vector $\mathscr{U} = (\mathbf{U}_1, \dots, \mathbf{U}_N)^t$, each Newton iteration for the computation of $\mathscr{U}$ at time step $n+1$ requires the numerical solution of the following linear system:

$$\mathscr{A}(\mathscr{U}^k)\delta\mathscr{U}^{k+1} = b(\mathscr{U}^n, \mathscr{U}^k). \tag{9}$$

Scaling strategy

The larger the time step, the worse the condition number of the matrix $\mathscr{A}$ in (9). As a consequence, it is important to apply a preconditioner before solving the linear system. The most popular choice is the Incomplete LU factorisation (later named ILU, see [1] for more details). The error made by the approximate factorisation using an ILU preconditioner depends on the size of the off diagonal coefficients of the matrix. For a better performance of the preconditioner, it is desirable that off diagonal entries of the matrix have small magnitudes.
Here, we use the Scaling strategy (see details in [3]) to improve the condition number of the matrix. This strategy is a similarity transformation. Combined with the classical ILU preconditioner this strategy has reduced significantly the GMRES iterations for local systems and the computational time.

4 Domain decomposition method

The object of the present work is to solve the compressible fluids by a nonoverlapping domain decomposition methods [13, 15, 11, 9], and more precisely by a Schur complement method. A simple attempt is to adapt the principle of the domain decomposition method for elliptic problems [14, 10] to our problems. As in the case of elliptic problems, the principle is that we decompose the global problem into independent subproblems which are solved by each processor. However, the implementation of these ideas in hyperbolic problems raise some technical difficulties such as:

- The scheme must be conservative.
- In the finite volume formulation, there is no unknown defined at the interface.
- The boundary condition of hyperbolic systems must depend on the characteristics of the problem.

Those difficulties are solved in [5] for the Euler equations by replacing the interface variables in the context of elliptic problems by the interface fluxes in the context

of hyperbolic problems. In this paper, we introduce a new interface variable which make the Schur complement method easy to build and allows us to treat diffusion terms.

Implicit Coupling

We recall the linear system at each Newton iteration of the implicit scheme (8):

$$\begin{aligned}\frac{\delta \mathbf{U}_i^{k+1}}{\Delta t} + & \sum_{j\in N(i)} \frac{s_{ij}}{v_i}\left[(A^- + D)(\mathbf{U}_i^k, \mathbf{U}_j^k)\right]\left(\delta \mathbf{U}_j^{k+1} - \delta \mathbf{U}_i^{k+1}\right) \\ = & -\frac{\mathbf{U}_i^k - \mathbf{U}_i^n}{\Delta t} - \sum_{j\in N(i)} \frac{s_{ij}}{v_i}\left[(A^- + D)(\mathbf{U}_i^k, \mathbf{U}_j^k)\right](\mathbf{U}_j^k - \mathbf{U}_i^k).\end{aligned}$$

We would like to solve (8) on N processors and each processor work on one subdomain. We see that it lacks $\delta \mathbf{U}_j^{k+1}$ to the computational unit of the subdomain I if the cell j belongs to another subdomain, and it is not calculable by the system since $\delta \mathbf{U}_j^{k+1}$ is to be calculated. Then the processor I needs from the processor J the value $\delta \mathbf{U}_j^{k+1}$ which is not yet available. Conversely, the processor J needs $\delta \mathbf{U}_i^{k+1}$ from the processor I.

A new interface variable

In order to include diffusion terms in the model and to use various schemes and various systems, we introduce a new interface flux variable $\delta\phi_{ij}$ (see [4]) at the domain interface between two neighboring cells C_i and C_j which belong to different subdomains:

$$\delta\phi_{ij} = \delta \mathbf{U}_j - \delta \mathbf{U}_i \tag{10}$$

In the case where the cell i of the subdomain I is at the boundary and has to communicate with the neighboring subdomains, we can rewrite the system (8) as:

$$\begin{aligned}\frac{\delta \mathbf{U}_i^{k+1}}{\Delta t} + & \sum_{j\in I, j\in N(i)} \frac{s_{ij}}{v_i}\left[(A^- + D)(\mathbf{U}_i^k, \mathbf{U}_j^k)\right]\left(\delta \mathbf{U}_j^{k+1} - \delta \mathbf{U}_i^{k+1}\right) \\ = & -\frac{\mathbf{U}_i^k - \mathbf{U}_i^n}{\Delta t} - \sum_{j\in N(i)} \frac{s_{ij}}{v_i}\left[(A^- + D)(\mathbf{U}_i^k, \mathbf{U}_j^k)\right](\mathbf{U}_j^k - \mathbf{U}_i^k) \\ & - \sum_{j\notin I, j\in N(i)} \frac{s_{ij}}{v_i}\left[(A^- + D)(\mathbf{U}_i^k, \mathbf{U}_j^k)\right]\delta\phi_{ij}^{k+1}\end{aligned}$$

We define $\mathbf{U}_I = (\mathbf{U}_1, \ldots, \mathbf{U}_m)^t$ the unknown vector of the subdomain I,

$$\delta\phi_I = (\delta\phi_{ij})_{i\in I, j\in J, j\in N(i)}, \tag{11}$$

$\mathscr{A}_I$ the local Neumann matrix of the subdomain I, and
$P_I = \sum_{j\notin I, j\in N(i)} \frac{s_{ij}}{v_i}\left[A^-(\mathbf{U}^k_{Roe}) + D(\mathbf{U}^k_{diff})\right]$, we can write the linear system as:

$$\mathscr{A}_I(\mathbf{U}^k)\delta\mathbf{U}_I^{k+1} = b_I(\mathbf{U}^n,\mathbf{U}^k) - P_I\delta\phi_I \tag{12}$$

By taking into account equations (10), (11) and (12),and denoting $\delta\Phi = (\delta\phi_I)$, $I = 1\dots N$ we can build an extended system that distinguishes the internal unknowns from the interface ones:

$$\left(\begin{array}{cccc|c} \mathscr{A}_1 & 0 & \dots & \dots & P_1 \\ 0 & \mathscr{A}_2 & 0 & \dots & P_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mathscr{A}_N & P_N \\ \hline M_1 & \dots & \dots & M_N & \mathbb{I} \end{array}\right)\left(\begin{array}{c}\delta\mathbf{U}_1\\ \delta\mathbf{U}_2\\ \dots\\ \delta\mathbf{U}_N\\ \delta\Phi\end{array}\right) = \left(\begin{array}{c} b_1\\ b_2\\ \dots\\ b_N\\ b_\Phi\end{array}\right) \tag{13}$$

where $\mathscr{A}_I$ is the matrix that couples the unknowns associated with internal cells of Ω_I whereas M_I links $\delta\mathbf{U}_I$ to $\delta\Phi$ through (10). Then, in our method, M_I comprises only 0 or ± 1.
The internal unknowns in (13) can be eliminated in favor of the interface ones to yield the following interface system:

$$S\delta\Phi = b_\Phi, \tag{14}$$

with $(S\delta\Phi) = \delta\Phi + \sum_{I=1}^N M_I{\mathscr{A}_I}^{-1}P_I\delta\phi_I$ and $(b_\Phi) = \sum_{I=1}^N M_I{A_I}^{-1}b_I$.
The computation of the matrix S is so costly as we have to inverse the local matrix $\mathscr{A}_I$. Fortunately, we do not have to compute explicitly the coefficients of S. All we need is to design the operator $\delta\Phi \to S\delta\Phi$. Then the equation (14) can be solved by, e.g., GMRES, BICGStab, or the Richardson methods. Once we solved the interface system, we know $\delta\Phi$ and then we can solve the internal unknowns on each processor using the equation (12).

5 Numerical Results

We have implemented our method for the compressible Navier-Stokes equations and the isentropic two-fluid model and compared the results obtained using single and multiple domains. After this validation, we compare the computation time of the ILU preconditioner, our method and our method with strategy Scaling ([3]).
Fig. 1 presents the computational time required to perform a time step of a fixed global problem of one million cells using upwind scheme. We compare the computational time required using the classical distributed method (red curve), the domain decomposition method (blue curve) and the domain decomposition method with scaling (green curve). We vary the number of processors up to 128. One can see that the domain decomposition method is comparable with classical distributed method and using scaling ([3]) is better.

Fig 3 shows the computational time required to perform the previous test but using

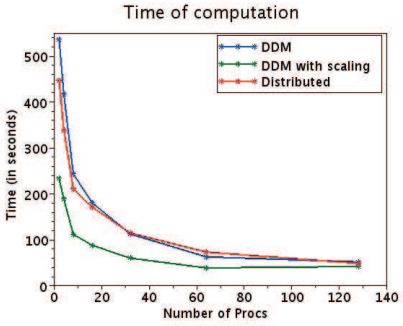


Fig. 1 Upwind scheme, single-phase flow, global mesh = $96 \times 96 \times 96$, CFL 20

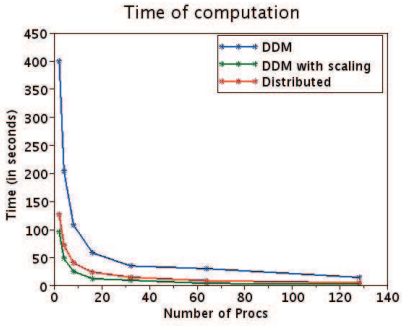


Fig. 2 Upwind scheme, two-phase flow, global mesh = 96x96x96, CFL 20

centered scheme. We can see only two curves. This is because, in this case the classical distributed method does not converge like we use the centered scheme. Domain decomposition is the only one method that converges.

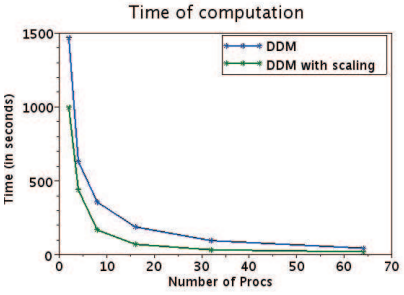


Fig. 3 Centered scheme, single-phase flow, global mesh = $96 \times 96 \times 96$, CFL 10

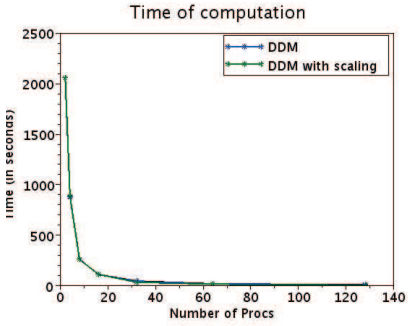


Fig. 4 Centered scheme, two-phase flow, global mesh = $96 \times 96 \times 96$, CFL 20

Similarly, Figs 2 and 4 show the computational time required to perform a time step in the case of the two-phase flow for the upwind and centered schemes.

Conclusion

We have presented a new interface variable which allows for the treatment of diffusion terms and the use of various numerical schemes for two-phase flows. We also introduced the Scaling strategy to improve the conditioner number of the matrix and reduce the computational time. We compared the scalability of our method with

the classical distributed computations. Numerical results showed that our method is more robust and efficient.

References

1. Benzi, M.: Preconditioning Techniques for Large Linear Systems: A Survey. J. Comput. Phys. **182** (2002)
2. Bergeaud, V., Fillion, P., Dérouillat, J.: Etude bibliographique sur l'inversion parallèle des matrices ovap. Tech. rep., Rapport CS/311-1/AB06A002-010/RAP/07-065 version 1.0 (2007)
3. Dao, T., Ndjinga, M., Magoulès, F.: Comparison of Upwind and Centered Schemes for Low Mach Number Flows. In: Proceedings of the International Symposium Finite Volumes for Complex Application VI, pp. 303–311. Springer Proceedings in Mathematics 4 (2011)
4. Dao, T., Ndjinga, M., Magoulès, F.: A Schur complement method for compressible Navier-Stokes equations. In: Proceedings of the 20th International Conference on Domain Decomposition Methods (2011)
5. Dolean, V., Lanteri, S.: A domain decomposition approach to finite volume solution of the Euler equations on unstructured triangular meshes. Int. J. Numer. Meth. Fluids **37**(6) (2001)
6. Drew, D.A., Passman, S.L.: Theory of Multicomponents Fluids. Springer, NY (1999)
7. Fillion, P., Chanoine, A., Dellacherie, S., Kumbaro, A.: Flica-ovap: a New Platform for Core Thermal-hydraulic Studies. In: NURETH-13 (2009)
8. Ishii, M.: Thermo-fluid Dynamic Theory of Two-phase Flow. Eyrolles, Paris (1975)
9. Kruis, J.: Domain Decomposition Methods for Distributed Computing. Saxe-Coburg Publications (2006)
10. Maday, Y., Magoulès, F.: Absorbing interface conditions for domain decomposition methods: a general presentation. Computer Methods in Applied Mechanics and Engineering **195**(29–32), 3880–3900 (2006)
11. Magoulès, F., Roux, F.X.: Lagrangian formulation of domain decomposition methods: a unified theory. Applied Mathematical Modelling **30**(7), 593–615 (2006)
12. Ndjinga, M., Kumbaro, A., Vuyst, F.D., Laurent-Gengoux, P.: Numerical simulation of hyperbolic two-phase flow models using a Roe-type solver. Nucl. Eng. Design **238** (2008)
13. Quarteroni, A., Valli, A.: Domain Decomposition Methods for Partial Differential Equations. Oxford University Press, USA (1999)
14. Smith, B., Bjorstad, P., Gropp, W.: Domain Decomposition : Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press, UK (1996)
15. Toselli, A., Widlund, O.B.: Domain Decomposition Methods - Algorithms and Theory. Springer (2005)
16. Toumi, I., Caruge, D.: An implicit second order method for 3d two-phase flow calculations. Nucl. Sci. Eng. **130** (1998)
17. Toumi, I., Kumbaro, A.: An Approximate Linearized Riemann Solver for a Two-Fluid Model. J. Comput. Phys. **124** (1999)

A Posteriori Error Estimates for a Neumann-Neuman Domain Decomposition Algorithm Applied to Contact Problems

Daniel Choï[1], Laurent Gallimard[2], and Taoufik Sassi[1]

1 Introduction

Contact problems are frequent in structural analysis. They are characterized by inequality constraints such as non-interpenetration conditions, sign condition on the normal constraints, and an active contact, an area that is a priori unknown. Several approaches exist for solving the non linear equations issued from the finite element discretization of frictionless contact problems. Recently, many efficient error estimates for solving frictionless contact problems have been proposed, see for example [1] and with domain decomposition techniques combined with adapative finite element methods, see [8, 5].

In this work, we consider a natural Neuman-Neumann domain decomposition (NNDD) algorithm, in which each iterative step consists of a Dirichlet problem for the one body, a contact problem for the other one and two Neumann problems to coordinate contact stresses. Two main approximation errors are introduced by this algorithm: a discretization error due to the finite element method (FEM) and an algebraic error due to the NNDD algorithm.

In [5] an error estimator in the constitutive relation for contact problems solved by a Neumann-Dirichlet domain decomposition algorithm has been proposed. The objective of this paper is to extend this error estimator for a frictionless contact problem, solved by a NNDD algorithm and to present two errors indicators which allow us to estimate the part of the error due to the spatial discretization and the part of the error due to the domain decomposition algorithm. Numerical results are presented, showing the pratical efficiency of the proposed error estimators.

2 A contact problem, notations and conventions

Two plane bounded domains Ω_1 and Ω_2 representing two linear elastic bodies are considered. Their Lipschitz boundaries are composed of distinct parts Γ_D^α, Γ_N^α and Γ_C^α :

$$\partial\Omega_\alpha = \overline{\Gamma_D^\alpha} \cup \overline{\Gamma_N^\alpha} \cup \overline{\Gamma_C^\alpha} \quad \alpha = 1,2.$$

[1] Laboratoire de Mathématiques Nicolas Oresme, Université de Caen Basse-Normandie, France, e-mail: {daniel.choi}{taoufik.sassi}@unicaen.fr .[2] Laboratoire Electronique, Mécanique, Energétique, Université Paris Ouest Nanterre-La Défense e-mail: laurent.gallimard@u-paris10.fr

The indices D, N, C of the boundary parts indicate respectively Dirichlet, Neumann and contact imposed boundary conditions, see problem (2)–(5). For the sake of simplicity, we suppose that $\Gamma_C^1 = \Gamma_C^2 = \partial\Omega_1 \cap \partial\Omega_2 = \Gamma_C$ is a common part of $\partial\Omega_\alpha$ along which the bodies Ω_α are in unilateral contact. On the presumed contact boundary Γ_C, we define

$$n = \quad n^1 = -n^2 \quad \text{and} \quad t = \quad t^1 = -t^2,$$

where n^α and t^α denote, respectively, the unit external normal and tangential vectors to $\partial\Omega_\alpha$.

On each domain Ω_α, $\alpha = 1,2$, the stress tensor is $\overline{\sigma}^\alpha$ and $\overline{\mathscr{E}}(u^\alpha)$ is the linearized strain tensor associated with the displacement u^α. With the elasticity tensors $\mathbb{E}^\alpha$, characterizing the materials of Ω_α, we have the linear strain-stress relation :

$$\overline{\sigma}^\alpha = \mathbb{E}^\alpha \overline{\mathscr{E}}(u^\alpha). \tag{1}$$

The bilinear energy forms, of linear elastic deformation, are then defined as

$$a^\alpha(u^\alpha, u^*) = \int_{\Omega^\alpha} \overline{\sigma}^\alpha : \overline{\mathscr{E}}(u^*).$$

The external loads (surfacic tractions of density F_α on Γ_N^α) are represented, in their weak form, as the linear forms b^α:

$$b^\alpha(u^*) = \int_{\Gamma_N^\alpha} F_\alpha . u^*.$$

3 Unilateral Contact problem and 'Neumann-Neumann' domain decomposition algorithm (NNDD)

We consider a unilateral frictionless contact problem between Ω_1 and With volumic forces neglected and tractions of density F^α imposed on Γ_N^α, the equilibrium equations can be written for $\alpha = 1,2$:

$$\mathrm{div}\overline{\sigma}^\alpha = 0 \quad \text{in } \Omega^\alpha \tag{2}$$

$$\overline{\sigma}^\alpha . n^\alpha = F^\alpha \quad \text{on } \Gamma_N^\alpha \tag{3}$$

with the kinematic boundary condition and unilateral frictionless contact conditions :

$$u^\alpha = u_D^\alpha \quad \text{on } \Gamma_D^\alpha \tag{4}$$

$$\left.\begin{array}{ll}(u^1-u^2).n & \leq 0\\ \sigma^1_{TN}=\sigma^2_{TN} & =0\\ \sigma^1_{NN}=\sigma^2_{NN} & =\sigma_N\\ \sigma_N & \leq 0\\ \sigma_N.(u^1-u^2).n & =0\end{array}\right\} \text{ on } \Gamma_C \tag{5}$$

with

$$\sigma^\alpha_{NN} = n^\alpha.\overline{\sigma}^\alpha n^\alpha \tag{6}$$
$$\sigma^\alpha_{NT} = t^\alpha.\overline{\sigma}^\alpha t^\alpha. \tag{7}$$

We now define a Neumann-Neumann domain decomposition (NNDD) algorithm. First, for any given normal displacement λ_p on Γ_C, we define the functional spaces

$$\begin{array}{ll}V^1 & =\{u\in H^1(\Omega^1); u|_{\Gamma^1_D}=u^1_D\}\\ U^1_C(\lambda_p) & =\{u\in V^1; u|_{\Gamma^1_C}.n=\lambda_p\}\\ V^2 & =\{u\in H^2(\Omega^2); u|_{\Gamma_D}=u^2_D\}\\ K^2_C(\lambda_p) & =\{u\in V^2; u|_{\Gamma^2_C}.n\geq\lambda_p\}.\end{array}$$

Given a non-negative parameter θ and an initial arbitrary λ_1, we define two sequences of displacements u^α_p on each solid Ω^α, $\alpha=1,2$. Each iteration p of the NNDD algorithm is divided in two successive steps.

- Step 1 – Two independent elasticity problems (hence parallelizable) are solved on Ω_1 and Ω_2:

  (i) In Ω_1, the variational problem writes

$$\left\{\begin{array}{l}\text{Find } u^1_p\in U^1_C(\lambda_p) \text{ such that}\\ a^1(u^1_p,u^*-u^1_p)=b^1(u^*-u^1_p)\quad \forall u^*\in U^1_C(\lambda_p)\end{array}\right. \tag{8}$$

  (ii) In Ω_2, with the given λ_p normal displacement defined on Γ_C, we solve the following variational problem corresponding to a unilateral frictionless contact problem on Γ^2_C :

$$\left\{\begin{array}{l}\text{Find } u^2_p\in K^2_C(\lambda_p) \text{ such that}\\ a^2(u^2_p,u^*-u^2_p)\geq b^2(u^*-u^2_p)\quad \forall u^*\in K^2_C(\lambda_p)\end{array}\right. \tag{9}$$

  From the respective unique solutions u^1_p and u^2_p of (8) and (9) we deduce r^1_p and r^2_p, defined on the contact Γ_C as

$$r^1_p=\overline{\sigma}^1_p n^1$$

$$r^2_p=\overline{\sigma}^2_p n^2.$$

where $\overline{\sigma}_p^1$ and $\overline{\sigma}_p^2$ are the stress tensor associated with the respective solutions u_p^1 and of u_p^2 of problems (8) and (9).

- Step 2 – With r_p^1 and r_p^2 obtained in step 1, we solve two independent "Neumann type" problems (hence the name NNDD):
  In Ω_1, we solve

$$\begin{cases} \text{Find } w_p^1 \in V^1 \text{ such that} \\ a^1(w_p^1, u^* - w_p^1) = -\int_{\Gamma_C} \frac{1}{2}(r_p^1 + r_p^2).(u^* - w_p^1) \quad \forall u^* \in V^1. \end{cases} \tag{10}$$

  In Ω_2, we solve

$$\begin{cases} \text{Find } w_p^2 \in V^2 \text{ such that} \\ a^2(w_p^2, u^* - w_p^2) = \quad \int_{\Gamma_C} \frac{1}{2}(r_p^1 + r_p^2).(u^* - w_p^2) \quad \forall u^* \in V^2. \end{cases} \tag{11}$$

  Let ε_t be the precision of the algorithm, we have the alternative :

  (i) If ε_τ is small enough, the algorithm stops.
  (ii) Else, the normal displacement λ_p is updated :

$$\lambda_{p+1} := \lambda_p + \theta(w_p^1 - w_p^2).n$$

  and we return to step 1 for iteration $p+1$.

If $r_p^1 + r_p^2 = 0$, it means that the equilibrium is satisfied on the contact interface, in other words the solutions u_p^1 and u_p^2 of step 1 constitute the unique solution of the reference problem (2)–(5). The proof of convergence of the NNDD algorithm (8)–(11) is given in [6] for any sufficiently small $\theta > 0$:

Theorem 1. *There is a $\theta_0 > 0$ such that for any $0 < \theta \leq \theta_0$, the NNDD algorithm for unilateral frictionless contact converges.*

4 Error estimates

The NNDD algorithm introduces two error sources. The first one is introduced by the solution of the FE problems (8)–(9). The second is introduced by the iterative NNDD algorithm. The global error is defined as the difference between the solution of the weak form of the reference problem u^α and the finite element solution computed from the NNDD algorithm u_h^α. Let

$$e_h = \sqrt{\sum_{\alpha=1}^{2} \|u^\alpha - u_h^\alpha\|_{u,\Omega^\alpha}^2} \text{ where } \|u\|_{u,\Omega^\alpha}^2 = \int_{\Omega^\alpha} \mathbb{E}^\alpha \overline{\mathscr{E}}(u) \,.\, \overline{\mathscr{E}}(u)\, d\Omega^\alpha$$

In the next section, we will define an a posteriori global error estimator, which is an adaptation to the NNDD algorithm of the error estimator proposed in [5], [4].

Moreover, we propose here two error indicators that allow us to estimate separately the part of the error due to the FE discretization and that due to the NNDD algorithm.

4.1 Global error estimator

The global error estimator is based on the concept of error in the constitutive relation [7]. Let us consider kinematically admissible displacements, i.e those satisfying (4), $\hat{v} = (v^1, v^2, v_N)$ and statically admissible stress tensor fields $\hat{c} = (\overline{\tau}^1, \overline{\tau}^2, t_c)$, i.e. those satisfying (5), where on Γ_c, with $w^\alpha = v^\alpha|_{\Gamma_c}$:

$$w_c = w^1 - w^2, \text{ and } t_c = \overline{\tau}^\alpha n^\alpha.$$

We define a global error estimator for any admissible $\hat{s} = (\hat{c}, \hat{v})$:

$$e_{CRE}(\hat{s}) = \left[\sum_{\alpha=1}^{2} \|\overline{\tau}^\alpha - \mathbb{E}^\alpha \overline{\mathscr{E}}(v^\alpha)\|^2_{\overline{\tau},\Omega^\alpha} + 2\int_{\Gamma_c} [\phi(-w_c) + \phi^*(t_c) + w_c.t_c]\, dS \right]^{1/2},$$

with

$$\|\overline{\tau}^\alpha\|^2_{\overline{\tau},\Omega^\alpha} = \int_{\Omega^\alpha} \overline{\tau}^\alpha : (\mathbb{E}^\alpha)^{-1}(\overline{\tau}^\alpha),$$

and where ϕ and ϕ^* are the conjugate convex potentials introduced in [2] to model the Coulomb's contstitutive law in a frictionless case:

$$\begin{aligned} \phi(v) &= \begin{cases} 0 & \text{if } v_N \geq 0 \\ +\infty & \text{otherwise} \end{cases} \\ \phi^*(t) &= \begin{cases} 0 & \text{if } t_N \leq 0 \text{ and } t_T = 0 \\ +\infty & \text{otherwise,} \end{cases} \end{aligned}$$

where the indices N and T indicate respectively the normal and the tangential component.
From [2, 3] the unilateral frictionless contact condition is equivalent to

$$\phi(-w_c) + \phi^*(t_c) + w_c.t_c = 0 \text{ on } \Gamma_C. \tag{12}$$

$e_{CRE}(\hat{s})$ is the constitutive relation error estimator for the admissible solution $\hat{s}$. It is equal to zero if and only if $\hat{s}$ is the exact solution of the unilateral frictionless contact problem (5)–(2). From [1], we have the upper bound,

$$e_{CRE}(\hat{s}) \geq e_h = \sqrt{\sum_{\alpha=1}^{2} \|u_h^\alpha - u^\alpha\|^2_{u,\Omega^\alpha}}.$$

4.2 Error indicators

The discretization error is estimated through a discretization error indicator computed for a second reference problem defined by (8)–(9) for a given λ_p. The only approximation used to solve this problem is the Finite Element approximation.

Let $\hat{s}_p = (\hat{u}_p, \hat{c}_p)$ be an admissible pair for this new reference problem, then the *discretization error* indicator is defined by

$$\eta_{h,p}^{dis} = e_{CRE}(\hat{s}_p).$$

To define an algorithm error indicator, we consider a third reference problem obtained with the Finite Element discretization of equations (2)–(5) (It is also necessary to introduce a discretized contact constitutive relations), the only approximation used to solve this problem is the Neuman-Neuman domain decomposition algorithm. Let $\hat{s}_h = (\hat{u}_h, \hat{c}_h)$ be an admissible pair for this third reference problem, then the *algorithm error* indicator is defined by

$$\eta_h^{NNDD} = e_{CRE}(\hat{s}_h)$$

To build the admissible fields $\hat{s}_p$ and $\hat{s}_h$, we use an adaptation of the techniques developed in [5].

5 Numerical results

We consider a test problem illustrating the reference problem (2)–(5). The domain Ω^1 is subject to a non-zero imposed displacement on a part Γ_D^1 of its boundary and to a rigid frictionless contact on another part $\Gamma_{D'}^1$. The domain Ω^2 has zero displacement imposed on Γ_D^2. Some surface forces F_N are imposed on Γ_N^1 to illustrate some loss of contact at the interface, see Figure 1. The two domains are in contact on Γ_C.

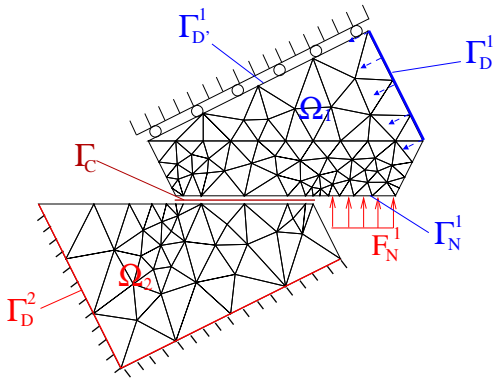


Fig. 1 A test problem for NNDD algorithm: frictionless unilateral contact between 2 elastic bodies.

In our implementation of the NNDD Algorithm, we define the precision of the algorithm ε_t as

$$\varepsilon_\tau = \frac{2\max_{\Gamma_C}|r_p^1 + r_p^2|}{\max_{\Gamma_C}|r_p^1| + \max_{\Gamma_C}|r_p^2|}$$

where r_p^1 and r_p^2 are obtained from step 1 of the NNDD algorithm at iteration p.

We first test the a posteriori error estimates of the NNDD algorithm (8)–(11) for different values of θ, and two meshes, one coarse mesh with 380 nodes and one finer mesh with 5994 nordes, see Figure 2. For both meshes, we notice an apparently optimal value near $0.4 \leq \theta \leq 0.5$ after 3 iterations of the NNDD algorithm. We also remark that the algorithm errors are very similar for both the fine and coarse meshes. The discretisation errors are naturally greater for the coarse mesh, but it doesn't change much with θ.

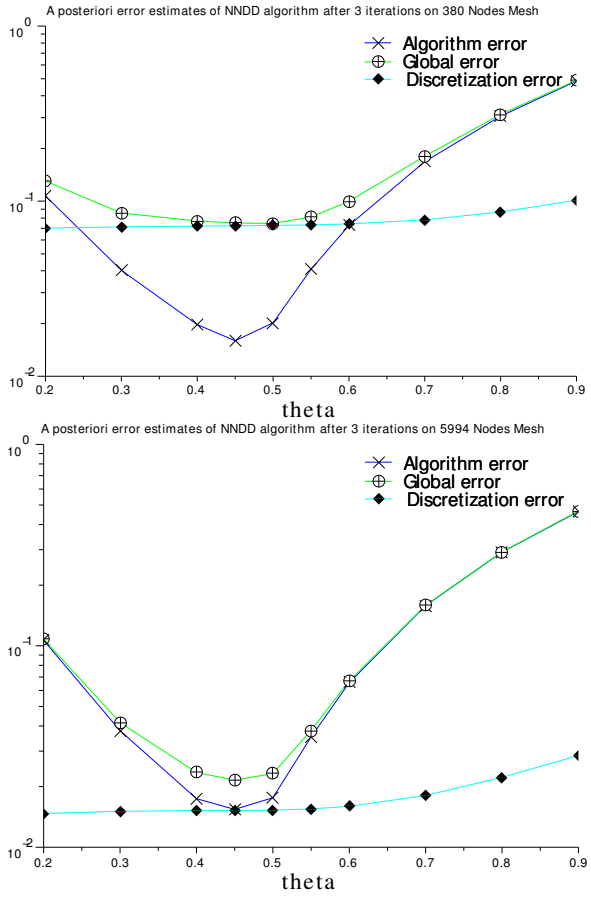


Fig. 2 NN error indicators for different values of θ, coarse 380 nodes mesh (up) and finer 5994 nodes mesh (down) after 3 iterations.

In Figure 3, we show the evolution of the algorithm error and the precision ε_t for an increasing number of iterations for a fixed value $\theta = 0.4$ and a fixed coarse mesh (380 nodes). While both decrease towards zero, the slopes of each appear very different. It means that the precision ε_t may not be a very good stopping criterion and can be deceiving as it appears much smaller than the algorithm error, which constitutes the largest part of the global error when using finer mesh, see the previuos figure 2.

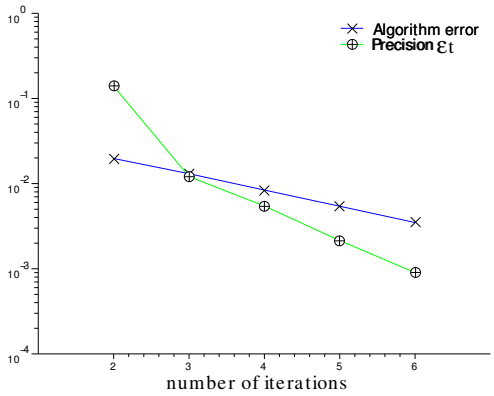


Fig. 3 Algorithm error and precision ε_t on fixed coarse mesh per number of iterations, with $\theta = 0.4$

References

1. Coorevits, P., Hild, P., Pelle, J.: Posteriori error estimation and indicators for contact problems. Comput. Methods Appl. Mech. Engrg. (2000)
2. De Saxcé, G.: A generalization of Fenchel's inequality and its applications to the constitutive laws. C. R. Acad. Sci. Paris Sér. II **314**(2), 125–129 (1992)
3. De Saxcé, G., Feng, Z.Q.: The bipotential method : a constructive approach to design the complete contact law with friction and improved numerical algorithms. Internat. J. Math. Comp. Model. **28**, 225–245 (1998)
4. Gallimard, L.: A constitutive relation error estimator based on traction-free recovery of the equilibrated stress. Internat. J. Numer. Engrg. (2009)
5. Gallimard, L., Sassi, T.: A posteriori error analysis of a domain decomposition algorithm for unilateral contact problem. Comput. & Structures **88**, 879–888 (2010)
6. Hasslinger, J., Kucera, R., Sassi, T.: A domain decomposition algorithm for contact problems: Analysis and implementation. Math. Model. Nat. Phenom. **4**(1), 123–146 (2009)
7. Ladevze, P., Leguillon, D.: Error estimate procedure in the finite element method and application. SIAM J. Numer. Anal. **20**, 485–509 (1983)
8. Pousin, J., Sassi, T.: A posteriori error estimates and domain decomposition with nonmatching grids. Adv. in Comp. Math **23**(241-263) (2005)

Additive Schwarz with Variable Weights

Chen Greif[1], Tyrone Rees[2], and Daniel B. Szyld[3]

1 Introduction and Motivation

We consider the numerical solution of nonsymmetric linear systems of equations of the form

$$A\mathbf{u} = \mathbf{f}, \tag{1}$$

that arise from the discretization of partial differential equations (PDEs). In practical problems, the number of mesh points is very large, and thus also the number of unknowns in (1), and the resulting matrix is large and sparse. In these circumstances, iterative methods are often used, due to their ability to deal more effectively with a high degree of sparsity. A popular iterative method is the *Generalized Minimum Residual* iterative scheme, or GMRES [8, 9, 10]. This method is based on minimizing at the kth iterate the residual within the affine Krylov subspace $\mathbf{u}_0 + \mathscr{K}^k(A, \mathbf{r}_0)$, where $\mathbf{u}_0$ is an initial vector, $\mathbf{r}_0 = \mathbf{f} - A\mathbf{u}_0$ is the initial residual, and

$$\mathscr{K}^k(A, \mathbf{r}_0) = \text{span}(\mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{k-1}\mathbf{r}_0).$$

The performance of GMRES is often (though not exclusively) determined by the structure of the eigenvalues of the matrix A. Loosely speaking, if they are strongly clustered, then GMRES is expected to converge fast. To accomplish a clustering effect, a *preconditioner* M is typically used: instead of solving (1) we solve, say,

$$AM\tilde{\mathbf{u}} = \mathbf{f},$$

where M is constructed so that AM has a more favorable eigenstructure than A. Upon incorporating the preconditioner M, the Krylov subspace changes accordingly: the matrix associated with the subspace becomes AM, and the preconditioned residual is now minimized.

A common way of dealing with the large number of degrees of freedom in a fine mesh is to break the problem down into a number of more manageable subproblems. This amounts to the technique of *domain decomposition*; see, e.g., [11]. We can then incorporate preconditioners that work on the subdomains into the general iterative framework.

The additive Schwarz preconditioner [11] and its restricted variant (RAS) [3], can be written in the form

[1] Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada. e-mail: greif@cs.ubc.ca .[2] Scientific Computing Department, STFC Rutherford Appleton Laboratory, Chilton, Didcot, UK. e-mail: tyrone.rees@stfc.ac.uk .[3] Department of Mathematics, Temple University, Philadelphia, Pennsylvania, USA. e-mail: szyld@temple.edu

$$M = \sum_{i=1}^{t} \tilde{R}_i A_i^{-1} R_i^T,$$

where t is usually the number of subdomains, $\tilde{R}_i$ is a restriction operator, R_i^T is a prolongation operator, and $A_i = R_i^T A R_i$ is the restriction of A onto the ith subdomain.

A possible generalization would be to use a weighted additive or restricted additive Schwarz preconditioner, say of the form

$$M^{(k)} = \sum_{i=1}^{t} \alpha_i^{(k)} \tilde{R}_i A_i^{-1} R_i^T,$$

where the weights $\alpha_i^{(k)}$ are chosen at the kth iteration of GMRES so as to minimize the preconditioned residual, cf. [1][1]. What we propose in this paper is to go a step further, and implicitly find at each iteration both the current weights and all the weights at the previous iterations, so as to minimize the residual at the current step.

Incorporating weights which change from one iteration to the next is significant and we can no longer talk about a standard iterative method with a single preconditioner. Instead, the proposed strategy fits into the MPGMRES paradigm the authors recently described in [5], where more than one preconditioner may be applied simultaneously.[2] Our main goal in this paper is to show that this methodology is particularly effective in the domain decomposition paradigm, since we can associate each subdomain with a specific, unique preconditioner.

An outline of the remainder of this paper follows. In Section 2 we briefly describe Additive and Restricted Additive Schwarz Preconditioning. In Section 3 we describe the MPGMRES algorithm. We address the question of computational cost of the algorithm and characterize the generalized Krylov subspace and its unique features in domain decomposition setting. In Section 4 we provide some details on numerical experiments. Finally, in Section 5 we make some concluding remarks.

2 Additive Schwarz Preconditioning

Suppose we divide the domain Ω containing n nodes into t subdomains $\Omega_1, \ldots, \Omega_t$, which overlap by bands of width δ nodes. Suppose each subdomain consists of $m_i \ll n$ nodes, which we denote as the entries of the set I_i. We can define a prolongation matrix $R_{i,\delta}^T \in \mathbb{R}^{n \times m_i}$ which extends vectors $\mathbf{u}^{(i)} \in \mathbb{R}^{m_i}$ to $\mathbb{R}^n$ by

$$(R_{i,\delta}^T \mathbf{u}^{(i)})_k = \begin{cases} (\mathbf{u}^{(i)})_k & \text{if } k \in I_i \\ 0 & \text{otherwise.} \end{cases}$$

[1] We point out that this is completely different than the approach in [4], where the weights are zeros and ones, and the emphasis is on asynchronous iterations.

[2] This algorithm extends previous work on using a combination of preconditioners – e.g., flexible GMRES [7] with alternating preconditioners, as described by Rui *et al.* [6] in the method they call multipreconditioned GMRES – by making an 'optimal' choice of weights. See [5] for a discussion.

The transpose of this matrix defines a restriction operator R_i which restricts vectors in $\mathbb{R}^n$ to the subdomain Ω_i. The restriction of the discretized PDE, A, to the ith subdomain is given by $A_i = R_{i,\delta} A R_{i,\delta}^T$.

We can now define the *additive Schwarz* preconditioner as

$$M := \sum_{i=1}^{t} R_{i,\delta}^T A_i^{-1} R_{i,\delta} = \sum_{i=1}^{t} M_i, \tag{2}$$

where $M_i := R_{i,\delta}^T (R_{i,\delta} A R_{i,\delta}^T)^{-1} R_{i,\delta}$. Note that, by the definition of $R_{i,\delta}^T$, there exists some permutation Π_i such that, for all $\mathbf{x}$,

$$\Pi_i M_i \mathbf{x} = (\times \cdots \times 0 \cdots\cdots 0)^T,$$

i.e., the vector resulting from multiplication by the M_i (regardless of the permutation) will be sparse.

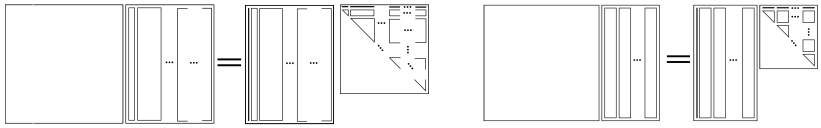
We can also define a *restricted additive Schwarz* (RAS) preconditioner [5] by considering the prolongation $R_{i,0}^T$ instead of $R_{i,\delta}^T$ in (2).

3 The MPGMRES Algorithm for Domain Decomposition Problems

MPGMRES [5] is a minimal residual algorithm for solving a linear system of equations which allows the user to apply more than one preconditioner simultaneously (see also [2] for a multipreconditioned version of the conjugate gradient method). At each step, new search directions are added to the search space, corresponding to $AM_i\mathbf{v}$ for each $i = 1,\ldots,t$, and for each basis vector $\mathbf{v}$ of the current search space. The multipreconditioned search directions are all combined into a generalized Krylov subspace, and the minimization procedure requires solving a linear least-squares problem. As opposed to standard GMRES, here the subspace grows quickly due to the presence of multiple search directions and the projection can be expressed in terms of a block upper Hessenberg matrix; see Figure 1. It has been shown in [5] that a so-called *selective MPGMRES* (sMPGMRES) algorithm – which chooses a subset of t search directions and hence keeps the size of the search space growing only linearly – can be an effective method. MPGMRES (in both complete and selective forms) is given as Algorithm 1.

3.1 Computational Work

In the selective algorithm we need t matrix-vector products and t preconditioner solves per iteration, as opposed to one for both in the standard preconditioned GMRES algorithm. The main other source for work is the inner products. Note



(a) Complete MPGMRES (b) Selective MPGMRES

Fig. 1 Schematic of Arnoldi decompositions in complete and selective MPGMRES

Algorithm 1 MPGMRES

Choose $\mathbf{u}_0$, $\mathbf{r}_0 = \mathbf{f} - \mathscr{A}\mathbf{u}_0$
$\beta = \|\mathbf{r}_0\|$, $\mathbf{v}_1 = \mathbf{r}_0/\beta$
$Z_1 = [\mathscr{M}_1\mathbf{v}_1 \cdots \mathscr{M}_t\mathbf{v}_1]$
for $k = 1, \ldots,$ **until** convergence **do**
  $W = \mathscr{A}Z_k$
  for $j = 1, \ldots, k$ **do**
    $H_{j,k} = (V_j)^T W$
    $W = W - V_j H_{j,k}$
  end for
  $W = V_{k+1}H_{k+1,k}$ (skinny QR factorization)
  $\mathbf{y}_k = \text{argmin}\|\beta\mathbf{e}_1 - \widetilde{H}_k\mathbf{y}\|_2$
  $\mathbf{u}_k = \mathbf{u}_0 + [Z_1 \cdots Z_k]\mathbf{y}_k$
  $Z_{k+1} = \begin{cases} [\mathscr{M}_1\mathtt{V}_{\mathtt{k+1}} \cdots \mathscr{M}_{\mathtt{t}}\mathtt{V}_{\mathtt{k+1}}] & \text{for complete MPGMRES} \\ [\mathscr{M}_1\mathtt{V}_{\mathtt{k+1}}\mathbf{1} \cdots \mathscr{M}_{\mathtt{t}}\mathtt{V}_{\mathtt{k+1}}\mathbf{1}] & \text{for selective MPGMRES} \end{cases}$
end for

that every entry in the Hessenberg matrix H_k is the result of an inner product, and these are the only inner products in the algorithm. MPGMRES therefore needs $(2k-1)\frac{t^2}{2} + \frac{3}{2}t$ inner products at the kth step [5, Table 4.1].

Significantly, in the domain decomposition setting, due to the nature of the standard Additive Schwarz preconditioner, the preconditioning step is *exactly* the same cost when using both selective MPGMRES and standard preconditioned GMRES. Moreover, since the vectors we obtain by applying the preconditioners are sparse, the cost of the matrix-vector products will also be of the same order as in the standard GMRES algorithm – the only extra expense coming from the overlapping nodes. Indeed, if we use RAS, then the cost of a matrix-vector product would be identical here too. While we studied RAS in the context of MPGMRES in [5], in the rest of this paper we restrict our comments and experiments to additive Schwarz. The extra cost in the MPGMRES approach therefore lies completely with the inner products. The vectors here are, in general, dense, as we lose sparsity of W in the modified Gram-Schmidt step (in the inner loop of Algorithm 1).

3.2 The subspace in complete MPGMRES

Recall that (complete) MPGMRES minimizes over the multi-Krylov subspace

$$\mathscr{K}^k_{M_1,\ldots,M_t}(A,\mathbf{r}_0),$$

where

$$\begin{aligned}\mathscr{K}^1_{M_1,\ldots,M_t}(A,\mathbf{r}_0) &= span\{M_1A\mathbf{r}_0,\ldots,M_tA\mathbf{r}_0\},\\ \mathscr{K}^2_{M_1,\ldots,M_t}(A,\mathbf{r}_0) &= span\{M_1A\mathbf{r}_0,\ldots,M_tA\mathbf{r}_0,M_1AM_1\mathbf{r}_0,\ldots,M_1AM_t\mathbf{r}_0,\ldots\\ &\qquad \ldots,M_tAM_1\mathbf{r}_0,\ldots,M_tAM_t\mathbf{r}_0\},\end{aligned}$$

etc. Usually the size of this space grows exponentially with each iteration. However, in an additive Schwarz context the situation is not quite so dire, as we see below.

First, note that each preconditioned matrix is a projection, since

$$M_iAM_i = R^T_{i,\delta}(R_{i,\delta}AR^T_{i,\delta})^{-1}R_{i,\delta}AR^T_{i,\delta}(R_{i,\delta}AR^T_{i,\delta})^{-1}R_{i,\delta} = M_i.$$

Hence applying M_i to AM_i does nothing to enrich the space.

Next, note that

$$M_iAM_j = R^T_{i,\delta}(R_{i,\delta}AR^T_{i,\delta})^{-1}R_{i,\delta}AR^T_{j,\delta}(R_{j,\delta}AR^T_{j,\delta})^{-1}R_{j,\delta}.$$

In the middle of this expression is the cross-term $R_{i,\delta}AR^T_{j,\delta}$. Now note that $R_{i,\delta}AR^T_{j,\delta} = 0$ whenever $I_i \cap I_j = \emptyset$. Provided the overlap δ is not large enough to touch two subdomains, this implies that only the contributions from sub-domains that touch each other add anything to the multi-Krylov subspace. This is the number of edges + corners in 2D (a maximum of 8 for a tensor product-based grid), and these plus the number of faces in 3D (a max of 26 for a tensor product-based grid). Altogether, this means that

$$dim(\mathscr{K}^k_{M_1,\ldots,M_t}(A,\mathbf{r}_0)) = (kc+1)t,$$

where c is a constant independent of k,t. Therefore, even in the complete MPGMRES case, we only have *linear* growth in the search space.

4 Numerical Experiments

If we split the domain into a small number of subdomains, i.e., we have a high proportion of subdomains lying on an edge, then there may not be much difference between the spaces minimized over by the selective algorithm and the complete algorithm.

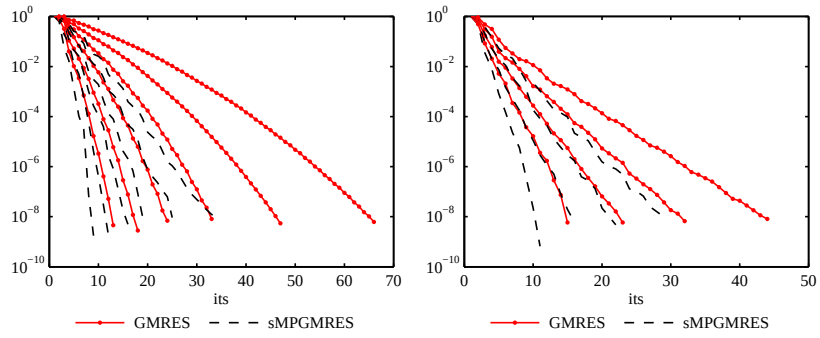
For example, consider the special case where we split the domain Ω into two subdomains, Ω_1 and Ω_2 such that $\Omega_1 \cup \Omega_2 = \Omega$. Then it can be shown [5, Section 5.2.1] that, provided the subdomain solves are exact, the space over which we minimize in both selective and complete MPGMRES are identical.

Figure 2 shows the convergence curves for solving the advection-diffusion equation

$$-\nabla^2 u + \omega \cdot \nabla u = f \qquad \text{in } \Omega \tag{3}$$

$$u = 0 \qquad \text{on } \partial\Omega, \tag{4}$$

where Ω denotes the unit square and $\omega = 10\left(\cos(\frac{\pi}{3}), \sin(\frac{\pi}{3})\right)^T$. This is discretized using finite differences with a uniform mesh size h, and the right hand side is taken to be the vector of ones. Thus, in 2D, $n = 1/h^2$ and in 3D, $n = 1/h^3$.



(a) 2D, $h = \{2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}, 2^{-8}\}$ (b) 3D, $h = \{2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}\}$

Fig. 2 Convergence curves for solving the advection-diffusion equation (3-4) with two subdomains in 2D and 3D. The iteration number is plotted along the x-axis, and $\|\mathbf{r}_k\|_2$ is plotted along the y-axis.

As we see in Figure 2, the iteration counts are significantly better using a multipreconditioned approach. Despite only having a serial MATLAB code, this also corresponds to significantly better timings, as is seen in Table 1: it is anticipated that the difference between the two approaches would be even more striking in a parallel implementation.

For a large numbers of subdomains, the work involved in the inner products and vector updates becomes significant, even though the work in actually applying the preconditioners is essentially the same as for the usual AS method. Convergence curves for the problem (3)-(4) are given in Figure 3.

Although the iteration counts are impressive for a large number of subdomains (with, e.g., 101 iterations for GMRES with an additive Schwarz preconditioner being reduced to 17 iterations with selective MPGMRES for 256 subdomains), the timings in this case are not yet competitive – e.g., for the case with 256 subdomains GMRES converges in 2.5s whereas sMPGMRES takes 9s. This is due to the fact

Table 1 Timings for sMPGMRES and GMRES with two subdomains in 2D (left) and 3D (right)

h	sMPGMRES	GMRES
2^{-3}	0.008	0.007
2^{-4}	0.015	0.023
2^{-5}	0.13	0.087
2^{-6}	0.32	0.55
2^{-7}	2.1	3.7
2^{-8}	15.3	28.6

h	sMPGMRES	GMRES
2^{-2}	0.010	0.011
2^{-3}	0.059	0.058
2^{-4}	1.03	1.49
2^{-5}	25.6	39.7

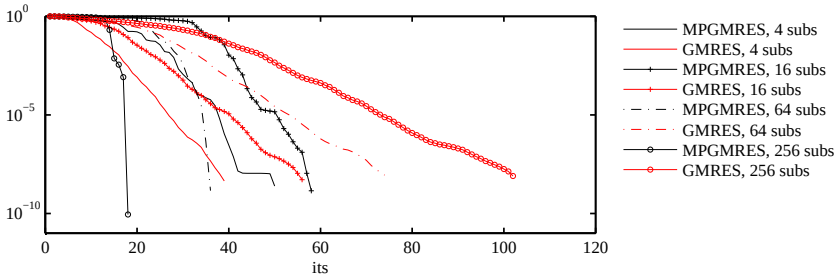


Fig. 3 Convergence curves for multiple subdomains in 2D ($h = 2^{-6}$). The iteration number is plotted along the x-axis, and $\|\mathbf{r}_k\|_2$ is plotted along the y-axis.

that we are using a proof-of-concept (serial) MATLAB code. Recall that the only extra work between the methods is in calculating the inner products and the subsequent vector update in the Gram-Schmidt process. Due to the block nature of the proposed method much of this extra work could be distributed across any available processors. We envisage that a state-of-the-art implementation would yield great computational savings, which would be manifested in a significantly reduced running time. This would be especially true for very large scale problems, where the cost of the subdomain solves would dominate the cost of each iteration. A Fortran 95 implementation of MPGMRES – `HSL_MI29` – will be included in the 2013 release of the HSL subroutine library.

Recall from Algorithm 1 that in the implementation of sMPGMRES reported here we apply each preconditioner to the sum of the columns of V_{k+1}. This choice is by no means unique, and there are many other possible selection strategies [5, Section 2.3]. The approach employed here seems to perform well on a wide range of problems, but it is a somewhat arbitrary choice. There may be situations where another selection strategy would be superior; this is one avenue for future research.

5 Conclusions

We have presented an algorithm that applies Additive Schwarz with Variable Weights. The approach is incorporated as a set of multiple preconditioners into MPGMRES. Domain decomposition has a few unique features that make our approach particularly attractive. First, the preconditioning step entails the same cost when using both selective MPGMRES and standard preconditioned GMRES, and the cost of the matrix-vector products is also of the same order as in the standard GMRES algorithm. Secondly, because there is a very low degree of overlap between nodes in the different subdomains, the growth in the search space for complete MPGMRES is only linear, i.e., very modest. This is in contrast to other situations, where the search space for complete MPGMRES grows exponentially and we settle for a selective algorithm. For these reasons we believe that the combination of domain decomposition preconditioners and the MPGMRES framework is an effective method for the numerical solution of linear systems arising from PDEs.

Acknowledgements The work of the first author was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and that of the third author in part by the U.S. National Science Foundation under grant DMS-1115520.

References

1. Ayuso de Dios, B., Baker, A.T., Vassilevski, P.S.: A combined preconditioning strategy for nonsymmetric systems (2012). Arxiv:1208.4544v1
2. Bridson, R., Greif, C.: A multipreconditioned conjugate gradient algorithm. SIAM Journal on Matrix Analysis and Applications **27**(4), 1056–1068 (2006)
3. Cai, X.C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM Journal on Scientific Computing **21**, 239–247 (1999)
4. Frommer, A., Schwandt, H., Szyld, D.B.: Asynchronous weighted additive Schwarz methods. Electronic Transactions on Numerical Analysis **5**, 48–61 (1997)
5. Greif, C., Rees, T., Szyld, D.B.: MPGMRES: a generalized minimum residual method with multiple preconditioners. Tech. Rep. 11-12-23, Department of Mathematics, Temple University (2011). Revised September 2012. Also available as Technical Report TR-2011-12, Department of Computer Science, University of British Columbia
6. Rui, P.L., Yong, H., Chen, R.S.: Multipreconditioned GMRES method for electromagnetic wave scattering problems. Microwave and Optical Technology Letters **50**(1), 150–152 (2008)
7. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. SIAM Journal on Scientific Computing **14**, 461–469 (1993)
8. Saad, Y.: Iterative methods for sparse linear systems, second edn. SIAM, Philadelphia (2003)
9. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on Scientific and Statistical Computing **7**, 856–869 (1986)
10. Simoncini, V., Szyld, D.B.: Recent computational developments in Krylov subspace methods for linear systems. Numerical Linear Algebra with Applications **14**, 1–59 (2007)
11. Toselli, A., Widlund, O.B.: Domain Decomposition Methods - Algorithms and Theory, *Springer Series in Computational Mathematics*, vol. 34. Springer, Berlin and Heidelberg (2005)

A parallel multigrid solver on a structured triangulation of a hexagonal domain

Kab Seok Kang[1]

1 Introduction

Fast elliptic solvers are a key ingredient of massively parallel Particle-in-Cell (PIC) and Vlasov simulation codes for fusion plasmas. This applies for both, the gyrokinetic and fully kinetic models. The currently available most efficient solver for large elliptic problems is the multigrid method, especially the geometric multigrid method which requires detailed information of the geometry for its discretization.

In this paper, we consider a structured triangulation of a hexagonal domain for an elliptic partial differential equation and its parallel solver. The matrix-vector multiplication is the key component of iterative methods such as CGM, GMRES, and the multigrid method. Many researchers have developed parallel solvers for partial differential equations on unstructured triangular meshes. In this paper, we consider a new approach to handle a structured grid of a regular hexagonal domain with regular triangle elements. We classify nodes as either real or ghost ones and find that the required steps of data communication to assign the values on the ghost nodes is five. We show that the matrix-vector multiplication of this approach has an almost perfect scaling property.

The multigrid method is a well-known, fast and efficient algorithm to solve many classes of problems [1, 4, 5]. In general, the ratio of the communication costs to computation costs increases when the grid level is decreased, i.e., the communication costs are high on the coarser levels in comparison to the computation costs. Since, the multiplicative multigrid algorithm is applied on each level, the bottleneck of the parallel multigrid lies on the coarser levels, including the exact solver at the coarsest level. The additive multigrid method could combine all the data communication for the different levels in one single step. However, this version can be used only for preconditioner and need almost the double amount of iterations generally. The multiplicative version can be used as a solver and as a preconditioner, so we consider the multiplicative version only.

The feasible coarsest level of operation of the parallel multigrid method depends on the number of cores. The number of degrees of freedom (DoF) of the coarsest level problem will be increased as the number of cores is increased. To improve the performance of the parallel multigrid method, we consider reducing the number of executing cores to one (the simplest case) after gathering data from all cores on a certain level. This algorithm avoids the coarsest level limitation and numerical experiments on large numbers of cores show very good performance improvement.

[1] Max-Planck-Institut für Plasmaphysik, EURATOM Associate, Boltzmannstraße 2, D-85748 Garching, Germany e-mail: kskang@ipp.mpg.de

A different way to improve the performance of the parallel multigrid method is to use a scalable solver on the coarsest level. A good candidate for the coarsest level solver is the two-level domain decomposition method because these methods are intrinsically parallel and their required number of iterations does not depend on the number of sub-domains (cores). We consider BDDC [2] and FETI-DP [3] because these are well-known two-level non-overlapping domain decomposition methods and show very good performance for many problems.

In this paper we investigate the scaling properties of the multigrid method with gathering data, BDDC, and FETI-DP on a massively parallel computer.

2 Model problem and its parallelization

We consider the Poisson type second order elliptic partial differential equations on a regular hexagonal domain Ω with Dirichlet boundary conditions

$$\begin{aligned} c(x,y)u - \nabla \cdot a(x,y)\nabla u &= f, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega, \end{aligned} \tag{1}$$

where $f \in L^2(\Omega)$, $c(x,y)$ is a non-negative function and $a(x,y)$ is a uniformly positive and bounded function. It is well known that the Eq. (1) has a unique solution.

The second-order elliptic problem (1) is equivalent to: find $u \in H_0^1(\Omega)$ such that

$$a_E(u,v) = \int_\Omega c(x,y)uv\,\mathrm{dx} + \int_\Omega a(x,y)\nabla u \cdot \nabla v\,\mathrm{dx} = \int_\Omega fv\,\mathrm{dx} \tag{2}$$

for any test function $v \in H_0^1(\Omega)$ where $H_0^1(\Omega)$ is the space of the first differentiable functions in Ω with zero values on the boundary $\partial\Omega$.

We consider a piecewise linear finite element space defined on a triangulation with regular triangles. This triangulation generate a structured grid and can be applied to a D-shape Tokamak interior region with conformal mapping. Let h_1 and $\mathscr{T}_{h_1} \equiv \mathscr{T}_1$ be given, where $\mathscr{T}_1$ is a partition of Ω into triangles and h_1 is the maximum diameter of the elements of $\mathscr{T}_1$. For each integer $1 < k \le J$, let $h_k = 2^{-(k-1)}h_1$ and the sequence of triangulations $\mathscr{T}_{h_k} \equiv \mathscr{T}_k$ be constructed by the nested-mesh subdivision method, i.e., let $\mathscr{T}_k$ be constructed by connecting the midpoints of the edges of the triangles in $\mathscr{T}_{k-1}$, and let $\mathscr{T}_{h_J} \equiv \mathscr{T}_J$ be the finest grid.

Let us define the piecewise linear finite element spaces

$$V_k = \{v \in C^0(\Omega) : v|_K \text{ is linear for all } K \in \mathscr{T}_k\}.$$

Then, the finite element discretization problem can be written as follows: find $u_J \in V_J$ such that

$$a_E(u_J, v) = \int_\Omega fv\,\mathrm{dx} \tag{3}$$

for any test function $v \in V_J$, i.e., solve the linear system $A_J u_J = f_J$.

Let us now consider the parallelization of the above problem. We use real and ghost nodes on each core. The values on the real nodes are handled and updated locally. The ghost nodes are the part of the distributed sub-domains located on other cores whose values are needed for the local calculations. Hence, the values of the ghost nodes are first updated by the cores to which they belong to as real nodes and

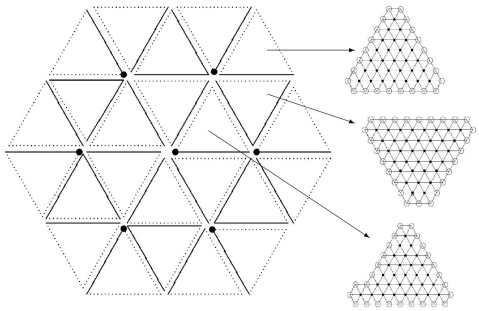


Fig. 1 The subdomains on 24 cores and real (–, •) and ghost (···, ◯) nodes on subdomains according to the types

then transferred to the cores that need them. To reduce data communication during matrix element computation, the computation of matrix elements on some cells can be executed on several cores which have a node of the cell as a real node.

We consider the way to divide the hexagonal domain into sub-domains with the same number of cores. Except for the single core case, we divide the hexagonal domain in regular triangular sub-domains and each core handles one sub-domain. Hence, feasible numbers of cores are limited to the numbers 6×4^n for $n = 0, 1, 2, \ldots$. For each core we have to define what are real and ghost nodes on the common boundary regions of the sub-domains. We determine the nodes on the common boundary of the sub-domains as the real nodes of the sub-domain which are located in the counterclockwise direction or in the outer direction from the center of the domain as shown in Fig. 1. For our problem with a Dirichlet boundary condition on the outer boundary, we can handle the boundary nodes as ghost ones. The values of these boundary nodes are determined by the boundary condition and thus do not have to be transferred bewteen cores.

We number the sub-domains beginning at the center and going outwards following the counterclockwise direction. Each sub-domain can be further divided into triangles; this process is called triangulation. In this process each line segment of the sub-domain is divided into 2^n parts. It can be shown that, independently of the total number of sub-domains and triangulation chosen, there are just three domain types. These give detailed information on the real and ghost nodes being connected to other sub-domains and cells which are needed to compute the matrix elements for the real nodes. To see how good the load balancing is, we measure the ratio of the largest number of real nodes to the smallest number of real nodes which is $\{2^n(2^n+3)\}/\{2^n(2^n+1)\}$ which tends to '1' as n is increased.

To get the values on the ghost nodes from the other cores for all sub-domains, we implement certain communication steps. The communication steps are the dominating part of the parallelization process and thus a key issue for the performance of the parallel code. The easiest way to implement the data communication would be that every ghost node value is received from the core which handles it as a real node value. However, such implementation would need several steps and the required number would then vary among the different cores. So this approach could be used for unstructured grids, but it would be too slow in our case. However, we solved the

problem by using a sophisticated data communication routine which needs a fixed number of steps for each core (that is, five).

Our dedicated data communication steps are as follows:

S1: Radial direction
S2: Counterclockwise rotational direction
S3: Clockwise rotational direction
S4: Radial direction (same as in S1)
S5: Mixed communications

3 Multigrid and domain decomposition methods

The motivation for the multigrid method is the fact that basic iterative methods, such as Jacobi and Gauss-Seidel methods, reduce well the high-frequency error but have difficulties to reduce the low-frequency error, which can be well approximated after projection on the coarser level problem. The multigrid method consists of two main steps, one is the smoothing operator and the other is the intergrid transfer operator. The former has to be easy to be implemented and be able to reduce effectively the high frequency error.

The other important operator is the intergrid transfer operator, which consists of the prolongation and the restriction operator. The intergrid transfer operators on triangular meshes have been studied in depth by many researchers, and their usage is mature.

The main issue with the parallelization of the multigrid method is execution time on the coarser level iterations. In general, the ratio of communication to computation on a coarse level grid is larger than on a fine level grid. Because the multigrid method works on both the coarse and fine grid levels, to get good scaling performance, we might need to avoid operating on the coarser level if possible. Usually, the W-cycle and the variable V-cycle multigrid methods require more work on the coarse level problems, so we consider for parallelization only the V-cycle multigrid method.

In addition to the execution time on the coarser level, we have to consider the solving time on the coarsest level. As a coarsest level solver, we can use either a Krylov subspace method or a direct method. The solving time of both methods increases with the problem size. So in considering the solution time of the coarsest level we need to find the optimal coarsening level, as well as the ratio of the communication to computation on each level.

From a certain level on, we can use a small number of cores to perform computations for the coarser levels. Among all the possible algorithms, let us consider the one which executes only on one core after having gathered all data.

Such a multigrid algorithm variation can solve the coarsest level problem on one core only, independent of the total number of cores. Instead of having only one core solving the coarser level problems and other cores idling, we choose to replicate the same computation on the coarser levels on each core; then we use these results for computations on the finer level. In the variant which we use, we use `MPI_Allreduce` which may yield a better performance than using combinations

of `MPI_Reduce` and `MPI_Bcast`, depending on the `MPI` implementation on the given machine.

Let us now consider another well known parallel solver, namely the domain decomposition method (DDM). The non-overlapping DDM is a natural method for problems which have discontinuous coefficients or many parts and are akin to being implemented on distributed memory computers. The non-overlapping DDM can be characterized by how it handles the values on the inner-boundary (that is, the common boundary of the two sub-domains). The condition number of the two-level non-overlapping DDM does not depend on the number of sub-domains. The BDDC and FETI-DP methods are well developed two-level DDM and have good performance when using a large number of sub-domains.

The BDDC algorithm [2] has been developed as an algorithm for substructuring, based on the constrained energy minimization concept. We follow the algorithm of [2] with a constraint matrix C_u which enforces equality of substructure DoF averaged across edges and at individual DoF on substructure boundaries (corner).

The FETI-DP method [3] imposes the continuity on the corner nodes which includes more than two sub-domains and the continuity on the edge nodes by using the Lagrange multipliers λ. By block Gauss elimination, we obtain the reduced system $F\lambda = d$ and solve it with PCGM with the Dirichlet preconditioner, as in [3].

4 Numerical experiments

As a model problem, we choose the simplest one with $c(x,y) = 0$ and $a(x,y) = 1.0$ in Eq. (1), i.e., the Poisson problem. To test the performance of our implementation, we use the finite element discretization formula which is the same for the finite volume discretization for this test problem. As a termination criterion for the solvers, we define a reduction of the initial residual error on the finest level by a factor of 10^{-8}.

The performance results reported in this paper were obtained on the HELIOS machine. The HELIOS machine is located in the International Fusion Energy Research Centre (IFERC) at Aomori, Japan. IFERC was built in the framework for the EU(F4E)-Japan broader approach collaboration. The machine is made by 4410 Bullx B510 Blades nodes of two 8-core Intel Sandy-Bridge EP 2.7 GHz processors with 64 GB memory and connected by Infiniband QDR. So it has a total of 70 560 cores total and 1.23 Petaflops Linpack performance.

We consider the multigrid method as a preconditioner of the preconditioned CGM with a localized Gauss-Seidel smoother which use old values on the ghost nodes. For the multigrid method, we use PCGM with the symmetric Gauss-Seidel method as a solver on the coarsest level and run two pre- and post-smoothing iterations for all cases.

We tested different data gathering levels on fixed numbers of cores. Without gathering data, the feasible coarsest level of the multigrid algorithm is the level that has at least more than one DoF per core. This level is the coarsest gathering level and depends on the number of cores. For instance, the coarsest gathering level of 384 cores is 5, of 1563 cores is 6, and 6144 cores is 7. Experimentally, setting the gathering level to the coarsest one shown always best performance. After gathering the data,

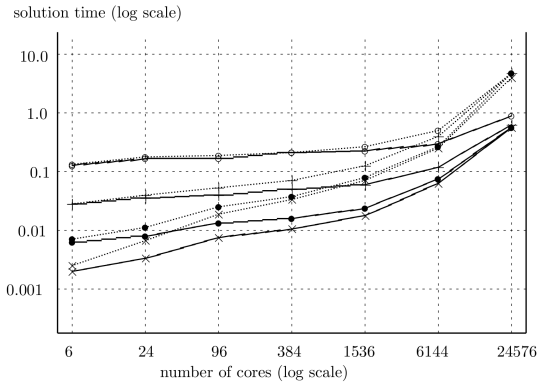


Fig. 2 The solution times in seconds of the multigrid method as a preconditioner for the PCGM with the Gauss-Seidel smoother with (–) and without (...) gathering data as a function of the number of cores for domains with 2.2K DoF (×), 8.5K DoF(•), 33.4K DoF(+), and 132K DoF(○) per core

all the computations are performed on one core. In this coarsest gathering level, the coarsest level does not impact performance as along as it is taken below level 6

In this paper, we use the simplest case only from the gathering level, all the data of the coarse problem are gathered on one core. In the case of large coarse problem, i.e., level greater than 6, a performance improvement could be expected by distributing it on many cores instead of one. But it has not been tested.

Let us now consider the performance impact when gathering the data on each core. To show that, we choose the coarsest level of the parallel algorithm as the coarsest gathering level. In the case of not gathering data, we have to use the coarsest gathering level as the coarsest level on which we solve the problem by using PCGM exactly. We tested four different cases, 2.2K, 8.5K, 33K, and 132K DoF per core and depicted the results in Fig. 2 which show that the gathering of the data is needed for large number of cores. The solution time of the solver in this case has a significant improvement for large numbers of cores and small number of DoF per core.

For a multigrid algorithm it is nearly impossible to fix the number of operations per core while increasing the total problem size, so we consider a semi-weak scaling by fixing the number of DoF of the finest level on each core. We tested six different number of DoF of the finest level on each core; from 2.2K DoF to 2.1M DoF and depicted the results in Table 1 together with the execution time (in bracket) of the matrix-vector multiplication which is the basic operation for iterative solvers and include the data communication step to update the values on the ghost nodes. The data shows that the matrix-vector multiplication has a perfect weak scaling property and the multigrid method as a preconditioner has really good semi-weak scaling properties when the number of DoF per core is large (compare 527K DoF and 2.1M DoF per core cases). Typically, the behaviour of multigrid algorithm implementations in weak scaling experiments is that they perform better as the number of DoF per core is increased.

The required number of iterations of the FETI-DP and BDDC methods does not depend on the number of sub-domains, but rather on the ratio of the mesh size of the triangulation (fine level, h) to the size of the sub-domains (coarse level, H). This

# cores	2.2K	8.5K	33.4K	132K	527K	2.1M
24	0.0034(0.000013)	0.0081(0.000055)	0.0356(0.00045)	0.1671(0.0031)	0.7046(0.0129)	2.824(0.052)
96	0.0075(0.000013)	0.0131(0.000056)	0.0406(0.00045)	0.1717(0.0031)	0.7114(0.0129)	2.825(0.051)
384	0.0104(0.000013)	0.0157(0.000056)	0.0502(0.00048)	0.2057(0.0031)	0.8397(0.0129)	3.327(0.052)
1536	0.0175(0.000013)	0.0244(0.000056)	0.0605(0.00051)	0.2209(0.0031)	0.8661(0.0129)	3.366(0.052)
6144	0.0633(0.000013)	0.0756(0.000056)	0.1192(0.00052)	0.3015(0.0031)	0.9476(0.0131)	3.471(0.052)
24576	0.5671(0.000014)	0.5630(0.000060)	0.6302(0.00054)	0.9105(0.0033)	1.6122(0.0141)	6.954(0.056)

Table 1 The solution times in seconds of the multigrid method as a preconditioner for the PCGM with the Gauss-Seidel smoother and the execution times of the matrix-vector multiplication (in bracket) according to the number of cores for domains with the several numbers of DoF per core

is shown in Table 2 where we list the required number of iterations of the FETI-DP and DBBC methods.

h/H	1/8		1/16		1/32		1/64		1/128	
# cores	FETIDP	BDDC	FETIDP	BDDC	FETIDP	BDDC	FETIDP	BDDC	FETIDP	BDDC
24	12	7	14	8	16	9	18	10	20	12
96	15	8	17	9	20	11	23	13	26	14
384	16	8	19	10	22	11	24	13	28	14
1536	16	8	20	10	23	11	26	13	29	14
6144	16	8	19	10	23	11	26	13	30	14
24576	16	8	19	9	23	11	26	13	29	14

Table 2 The required number of iterations of FETI-DP and BDDC according to the number of sub-domains and the ratio of the mesh size of the fine level (h) to the coarse level (H)

To implement the FETI-DP and BDDC methods, we have to solve local problems with Dirichlet and/or Neumann boundary conditions on each sub-domain and one globally defined coarse level problem. Furthermore, we need to communicate data with neighboring sub-domains and data on the coarse level. Solving the local problems and communicating the data with neighboring sub-domains are performed in parallel. So, these local steps do not alter the performance by changing the number of cores. Otherwise, the dimension of the global coarse level problem would grow as the number of cores increases. The dimension of the coarse level problem used for BDDC method is the same as of the coarsest gathering level used for the multigrid method. And the dimension of the coarse level problem used for FETI-DP method is one level below it.

We use the same gathering algorithm as in the multigrid method to solve the global coarser level problem. In both FETI-DP and BDDC, every sub-domain has some contributions to the matrices and vectors on the coarse level and uses the solution of the coarse level problem. So, we gather these contributions on each core using the `MPI_Allreduce` and use the solution after solving the coarse problem without any data communication.

To solve the local and global problems, we used two direct methods, the LAPACK (Intel MKL) library with dense matrix format and the IBM WSMP library with sparse matrix format, and the multigrid method as an iterative method. For large number of cores (more than 1536 cores), the global problems could be solved by either the iterative method or parallelized direct methods only on a small number of cores, due to the memory limitation. The solution time with parallel solver for

the global problems could be reduced as same as progressively reduced cores on the multigrid method.

For comparison to our previous results, we chose the solver which performes best. We tested five different cases with fixed number of DoF per core, from 55 DoF to 2200 DoF, and depicted the results in Table 3 together with the multigrid method. Results in Table 3 show that the FETI-DP is faster than the latter even though the DBBC requires a smaller number of iterations. These results also show that the weak scaling property is improved as the number of DoF per core is increased.

DoF/core	55			170			590			2200		
# cores	MG	FETI-DP	BDDC	MG	FETI-DP	BDDC	MG	FETI-DP	BDDC	MG	FETI-DP	BDDC
24	0.0009	0.0013	0.0014	0.0015	0.0020	0.0027	0.0019	0.0115	0.0216	0.0034	0.1007	0.2328
96	0.0022	0.0024	0.0028	0.0038	0.0034	0.0046	0.0054	0.0165	0.0298	0.0075	0.1287	0.3309
384	0.0043	0.0041	0.0067	0.0065	0.0057	0.0181	0.0080	0.0228	0.0439	0.0104	0.1414	0.3513
1536	0.0126	0.0131	0.0171	0.0152	0.0240	0.0367	0.0146	0.0512	0.0666	0.0175	0.1953	0.4056
6144	0.0582	0.0792	0.2954	0.0550	0.0988	0.3809	0.0584	0.1509	0.4849	0.0632	0.3864	1.2242
24576	0.5550	0.4961	1.8470	0.5762	0.5359	2.3163	0.5505	0.6883	2.3867	0.5671	1.0609	3.7620

Table 3 The solution times in seconds of the FETI-DP, the BDDC, and the multigrid method (MG) as a function of the number of cores for domains with the number of DoF per core

The solution times of the FETI-DP and the multigrid method for the smallest number of DoF per core cases (55 DoF per core) are almost the same. The multigrid method with gathering data is faster than the FETI-DP method. The difference of the solution time between the two methods increases as the number of DoF per core is increased, except for the largest number of cores (24576 cores).

5 Conclusions

We investigated the performance of the multigrid method with gathering data, BDDC, and FETI-DP on a regular hexagonal domain with regular triangulations and concluded that the first is the fastest solver for such a problem.

Acknowledgements This work was carried out using the HELIOS supercomputer system at Computational Simulation Centre of International Fusion Energy Research Centre (IFERC-CSC), Aomori, Japan, under the Broader Approach collaboration between Euratom and Japan, implemented by Fusion for Energy and JAEA. I would like to thank R. Hatzky and other HLST team members, B. Scott, and D. Tskhakaya for helpful discussions.

References

1. Bramble, J.: Multigrid Methods. Pitman, London (1993)
2. Dohrmann, C.R.: A preconditioner for substructuring based on constrained energy minimizations. SIAM J. Sci. Comput. **25**, 246–258 (2003)
3. Farhat C. Lesoinne M., e.a.: FETI-DP: A dual-primal unified FETI method – part I: A faster alternative to the two-level FETI method. Internat. J. Numer. Methods Engrg. **42**, 1523–1544 (2001)
4. Hackbush, W.: Multigrid Methods and Applications. Spinger-Verlag, Berlin (1985)
5. Hülsemann F. Kowarschik M., e.a.: Parallel geometric multigrid. Numerical Solution of Partial Differential Equations on Parallel Computers II, Lecture Notes in Computational Science and Engineering **51**, 165–208 (2006)

A parallel Crank–Nicolson predictor-corrector method for many subdomains

Felix Kwok[1]

1 Introduction

In this paper, we propose a fast parallel solver for the parabolic equation

$$\begin{aligned} \partial_t u &= \mathscr{L}u + g(x,t), \qquad x \in \Omega, \\ u = u_\Gamma(x,t) \quad &\text{on } \Gamma = \partial\Omega, \qquad u(x,0) = f(x) \quad \text{on } \Omega, \end{aligned} \tag{1}$$

where Ω is an open connected subset of $\mathbb{R}^2$ and $\mathscr{L}u = \sum_{i,j} \partial_{x_i}\left(\kappa_{ij}(x)\partial_{x_j}u\right) - c(x)u$, with $c(x) \geq 0$ and $\kappa_{ij}(x)$ symmetric and uniformly positive definite, i.e., we have $\kappa_{ij}(x) = \kappa_{ji}(x)$ for $i \neq j$ and $\sum_{i,j} \kappa_{ij}(x)\xi_i\xi_j \geq \lambda \sum_i \xi_i^2$ for all choices of ξ_i, where the constant $\lambda > 0$ is independent of x. Our method is based on the predictor-corrector method introduced by [15]. In that work, the authors consider nonlinear reaction-diffusion equations posed on branched structures, which model the evolution of the electric potential in neurons, see Fig. 1. In such problems, the nodal points are natural separators of the computational domain, meaning that the solution within the individual branches can be solved independently if the electric potential at the nodes are known. Based on this observation, the authors proposed the Crank–Nicolson predictor-corrector (CNPC) method: they first use forward Euler to predict the nodal values, and then backward Euler to solve for the solution within the branches. To maintain stability, they then correct the nodal values using a backward Euler step, and the whole solution is extrapolated to obtain formal second-order accuracy in time. The main advantage of this method is that a fixed amount of computation is performed at each time step, and no iteration is necessary. This is unlike classical domain decomposition (DD) algorithms such as Schwarz methods [3, 14, 11, 1] or waveform relaxation methods [10, 8, 9, 7], where one must iterate to convergence (or to some fixed tolerance), and the number of iterations generally increases as the grid is refined. Thus, a suitable extension of the CNPC method for 2D and 3D problems can be useful for parallel-in-time methods such as Parareal [13, 6], where fast coarse integrators are needed. Other DD-type methods with a fixed cost per time step have been proposed in [4] and [16]; both are only first order accurate under simultaneous refinement in space and time.

Our main goal is to present in detail a generalization of the CNPC method that can be used to solve 2D problems with many subdomains in parallel. This is done in Section 2. In particular, we show how the backward Euler correction step for the interface can be implemented efficiently, even in cases where the subdomain inter-

[1] Université de Genève, 2-4 Rue du Lièvre, 1211 Genève, Switzerland, e-mail: felix.kwok@unige.ch

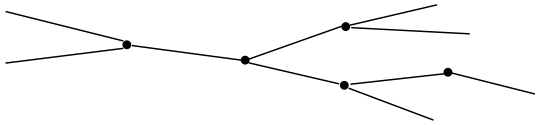


Fig. 1 A branched structure, with nodes indicated.

faces are coupled through cross points. To fix ideas, we have chosen a finite volume discretization in space, although similar techniques can be used for other discretizations. In Section 3, we examine the convergence of the CNPC method. We will see that the method indeed converges as the mesh size $h \to 0$ the time step τ satisfies $\tau = O(h^\alpha)$ for $\alpha \geq 1$. In fact, the method attains full second order accuracy for $\alpha \geq 3/2$; it is however only first order accurate when $\tau = O(h)$. Finally, numerical results in Section 4 illustrate the behavior of the method for many subdomains.

2 The CNPC algorithm

To define the CNPC algorithm, we will assume that the domain Ω is divided into shape regular, quasi-uniform and conforming control volumes V_i, $i = 1,\dots,n$, with diameter $h_i \leq h$, see Fig. 2. If we discretize (1) in space using a finite volume method, we get a semi-discrete ODE system of the form

$$M\partial_t u(t) + Au(t) + Bu_\Gamma(t) = Mg(\cdot,t). \tag{2}$$

Here, $u(t)$ are the unknown values at the nodal points at time t, $A \in \mathbb{R}^{n\times n}$ is a sparse, symmetric positive definite matrix whose entries a_{ij} are non-zero and of $O(1)$ (constant with respect to h) if and only if volumes i and j are neighbors. $B \in \mathbb{R}^{n\times n_\Gamma}$ contains the dependence on the Dirichlet boundary values; its entries are also $O(1)$. $u_\Gamma(t) \in \mathbb{R}^{n_\Gamma}$ contains the Dirichlet boundary values at time t. M is a diagonal mass matrix whose (i,i) entry is the area of V_i; thus, the elements of M are of size $O(h^2)$. $g(\cdot,t)$ is a vector whose elements are the values of g at the nodes; we will use this dot notation to denote the vectors of samples of other functions elsewhere in this paper.

We now divide the unknowns into two subsets, the *interface unknowns* $\mathcal{V}_1$ and the *interior unknowns* $\mathcal{V}_2$. We also define two corresponding projectors $X_1, X_2 \in \mathbb{R}^{n\times n}$ such that $X_1 u$ projects onto $\mathcal{V}_1$, i.e., it leaves all the values in $\mathcal{V}_1$ unchanged and sets all the other entries to zero, and X_2 does the opposite. Thus, we have $X_2 = I - X_1$ and $X_1X_2 = X_2X_1 = 0$. Note that X_1 and X_2 commute with M, since the latter is diagonal.

We are now ready to define the CNPC algorithm. For a given time-step size τ and an approximation $u^n \approx u(\cdot,t_n)$, one step of the CNPC method proceeds as follows:

(i) Predict the interface values at $t = t_{n+1/2}$ using forward Euler: calculate u^* using

$$\frac{M(u^* - u^n)}{\tau/2} = -X_1(Au^n + Bu_\Gamma(t_n)) + X_1 Mg(\cdot,t_{n+1/2}),$$

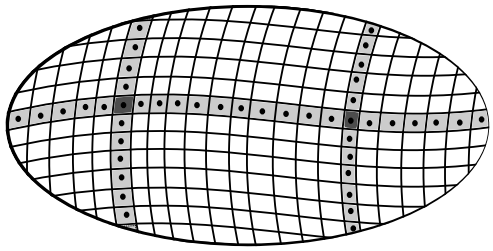


Fig. 2 Decomposition into interface (light and dark gray) and interior (white) cells and their corresponding unknowns. Light gray corresponds to edge nodes and dark gray to cross points.

Note that $X_2(u^* - u^n) = 0$, so interior node values are not altered by this step.

(ii) Using the predicted values $X_1 u^*$ as boundary values, solve for u^{**} in

$$\frac{M(u^{**} - u^n)}{\tau/2} = -X_2 \left[A(X_1 u^* + X_2 u^{**}) + B\Big(\frac{u_\Gamma(t_n) + u_\Gamma(t_{n+1})}{2} \Big) \right] + X_2 M g(\cdot, t_{n+1/2}),$$

where both $u_\Gamma(t_n)$ and $u_\Gamma(t_{n+1})$ are known. This corresponds to a backward Euler step for the interior unknowns $\mathcal{V}_2$; the interface values are not updated. Note that this step requires solving a linear system with the matrix $M + \frac{\tau}{2} X_2 A X_2$.

(iii) Compute $u^{n+1/2}$ by correcting the interface values at $t = t_{n+1/2}$ with backward Euler, using u^{**} as boundary values:

$$\frac{M(u^{n+1/2} - u^{**})}{\tau/2} = -X_1 \left[A(X_1 u^{n+1/2} + X_2 u^{**}) + B\Big(\frac{u_\Gamma(t_n) + u_\Gamma(t_{n+1})}{2} \Big) \right] + X_1 M g(\cdot, t_{n+1/2}).$$

This is a backward Euler step for the interface nodes, since their values have not been updated in the previous steps, i.e., we have $X_1 u^{**} = X_1 u^n$. For the other nodes, we have $X_2 u^{n+1/2} = X_2 u^{**}$, i.e. we reproduce the values obtained in step 2. Here one needs to solve a linear system with matrix $M + \frac{\tau}{2} X_1 A X_1$.

(iv) Extrapolate to obtain u^{n+1}:

$$u^{n+1} = 2u^{n+1/2} - u^n.$$

Note that there is no iteration to convergence, since each of the above is only performed once per time step.

Parallelization. We only need consider how to solve linear systems with matrices $A_i = M + \frac{\tau}{2} X_i A X_i$ ($i = 1, 2$) in parallel, since the other operators are local in nature and easy to parallelize. For the matrix $A_2 = M + \frac{\tau}{2} X_2 A X_2$ (step 2), we note that the interior nodes $\mathcal{V}_2$ are naturally decomposed into disconnected "subdomains" whose only connections are through the interface nodes $\mathcal{V}_1$. Thus, A_2 is block diagonal, with blocks corresponding to subdomains or to individual nodes in $\mathcal{V}_1$. As a result, if we assign each subdomain to its own processor, step 2 can be solved in parallel.

Next, we need to solve systems involving $A_1 = M + \frac{\tau}{2} X_1 A X_1$ (step 3). This is a block diagonal matrix whose largest block is of the same size as $\mathscr{V}_1$, so it is much smaller than the original system. Also note that $X_1 A X_1$ (and hence A_1) is *sparse*, with nonzero entries corresponding to neighboring interface nodes *only*. This is unlike a Schur complement approach, where the elimination of interior nodes introduces additional connections between non-neighboring interface nodes. However, the unknowns corresponding to edges from different subdomains are coupled through cross points, see Fig. 2, leading to a system that is globally coupled.

We now show how we can overcome this bottleneck by reducing the interface system to an even smaller one that has only as many variables as there are *cross points* in the domain. Let N be the number of subdomains, i.e., the number of connected components of $\mathscr{V}_2$. We partition the set $\mathscr{V}_1$ of interface nodes into edges $\{\mathscr{E}_1, \dots, \mathscr{E}_m\}$ between subdomains and $\mathscr{C}$, the set of cross points, so that $\mathscr{V}_1 = \mathscr{C} \cup \left(\cup_{j=1}^m \mathscr{E}_j\right)$. We now permute the blocks of A_1 so that edges are ordered first and the cross points last. If we let u_j be the unknowns corresponding to $\mathscr{E}_j$ and v be those belonging to cross points, we get

$$\begin{bmatrix} E_1 & & & G_1 \\ & E_2 & & G_2 \\ & & \ddots & \vdots \\ & & E_m & G_m \\ G_1^T & G_2^T & \cdots & G_m^T & C \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \\ v \end{pmatrix} = \sum_{i=1}^N f_i,$$

where E_j are sparse matrices corresponding to couplings within $\mathscr{E}_j$, G_j are the connections between $\mathscr{E}_j$ and the cross points, and C represents the connections among cross points themselves (typically $C = 0$). The f_i represent contributions of subdomain i to the right-hand side, e.g., contributions from nodes in subdomain i that are adjacent to $\mathscr{E}_j$. Then the Schur complement with respect to the cross points becomes

$$\left(C - \sum_{j=1}^m G_j^T E_j^{-1} G_j\right) v = R_C \sum_{i=1}^N \sum_{j=1}^m (I - R_j^T G_j^T E_j^{-1} R_j) f_i, \tag{3}$$

where R_j is the restriction from $\mathscr{V}_1$ to $\mathscr{E}_j$, $j = 1, \dots, m$ and R_C the restriction from $\mathscr{V}_1$ to $\mathscr{C}$.Thus, each term in the sum on the right-hand side can be computed independently by subdomain i; moreover, since edges are one-dimensional, E_j is typically a tridiagonal matrix that can be factored easily. In addition, $R_j f_i$ is nonzero only if $\mathscr{E}_j$ is an edge of subdomain i, so the inner sum contains only as many terms as there are edges in the subdomain boundary. Thus, the contribution $G_j^T E_j^{-1} G_j$ and the corresponding right-hand side can be calculated in parallel, and it remains to solve the Schur complement system, whose size is typically comparable to the number of subdomains. Once v is known, the u_j can be calculated in parallel by back substitution, which completes Step 3 in the CNPC algorithm. Thus, the cost of the coarse solve is low, similar to the cost of one coarse grid correction step in other domain decomposition methods, such as FETI-DP [5].

3 Convergence of the CNPC method

In this section, we outline the convergence analysis of the CNPC method under simultaneous time and spatial grid refinement. For more details, see [12]. For ease of presentation, we assume a uniform rectangular grid in which all control volumes are of size h^2, so that $M = h^2 I$. Then (2) is a second-order discretization of (1):

$$-\mathscr{L}u(\cdot,t) = \frac{1}{h^2}[Au(\cdot,t) + Bu_\Gamma(t)] + O(h^2).$$

We assume that the boundary data and source terms are sufficiently smooth, so that $u(x,t)$ has as many continuous spatial and temporal derivatives as needed.

Lemma 1. *The CNPC method can be written as*

$$Du^{n+1} + \frac{k}{2}(I + \frac{k}{2}X_2AX_1)Bu_\Gamma(t_{n+1}) = Cu^n - \frac{k}{2}(I - \frac{k}{2}X_2AX_1)Bu_\Gamma(t_n) + \tau g(\cdot, t_{n+1/2}),$$

where $k = \tau/h^2$, $D = (I + \frac{k}{2}X_2A)(I + \frac{k}{2}X_1A)$ and $C = (I - \frac{k}{2}X_2A)(I - \frac{k}{2}X_1A)$. Moreover, the stability matrix $D^{-1}C$ satisfies $\|D^{-1}C\|_W < 1$ for any $\tau > 0$ and $h > 0$, where $\|\cdot\|_W$ is induced by the vector norm $\|u\|_W^2 := u^T(I + \frac{k}{2}AX_1)A(I + \frac{k}{2}X_1A)u$.

Recall that the classical Crank–Nicolson method can be written as

$$(I + \frac{k}{2}A)u^{n+1} + \frac{k}{2}Bu_\Gamma(t_{n+1}) = (I - \frac{k}{2}A)u^n - \frac{k}{2}Bu_\Gamma(t_n) + \tau g(\cdot, t_{n+1/2}).$$

Thus, we see that CNPC and the classical Crank–Nicolson (CN) method differ by

$$\hat{\rho}_n := \frac{k^2}{4}X_2AX_1[A(u^{n+1} - u^n) + B(u_\Gamma(t_{n+1}) - u(t_n))] = -\frac{\tau^3}{4h^2}X_2AX_1\left(\mathscr{L}(\partial_t u(\cdot, t_{n+1/2})) + O(h^2)\right).$$

This observation, combined with the fact that the truncation error of CN is $O(\tau^2 + h^2)$, yields the following lemma.

Lemma 2. *The local truncation error ρ_n of the CNPC method at time step n satisfies*

$$\rho_n = \tau\left[-\frac{\tau^2}{4h^2}X_2AX_1\left(\mathscr{L}(\partial_t u(\cdot, t_{n+1/2})) + O(\tau^2) + O(h^2)\right) + O(\tau^2) + O(h^2)\right].$$

In particular, if $\tau = O(h^\alpha)$ with $\alpha \geq 1$, then $\rho_n = \tau \cdot [O(h^2) + O(h^{2\alpha-2})]$.

Note that the $O(h^{2\alpha-2})$ term comes from the term $\frac{\tau^2}{4h^2}X_2AX_1$. Fig. 3 shows the local truncation error for a two-subdomain decomposition with $\tau = O(h)$, for which Lemma 2 predicts $\rho_n/\tau = O(1)$. Although this is true near the interface, we observe that the error is much smaller away from the interface, where X_2AX_1 vanishes.

Let $\varepsilon_n := u(\cdot, t_n) - u^n$ denote the global error of the method at step n. If $\varepsilon_0 = 0$, i.e., if the correct initial conditions are used, then a standard argument shows that

$$\varepsilon_n = \sum_{j=1}^{n}(D^{-1}C)^{n-j}D^{-1}\rho_{j-1}.$$

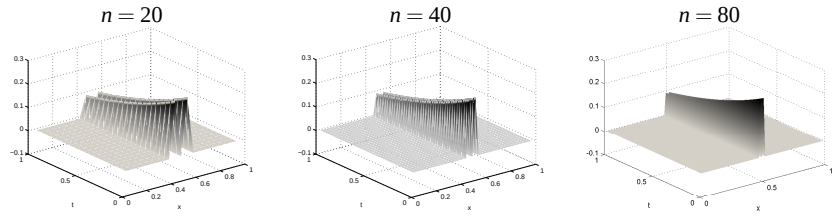


Fig. 3 Local truncation error of the CNPC method for a 1D two-subdomain problem with $u_t = u_{xx} + g(x,t)$, $\tau = h = 1/n$, where $n = 20, 40, 80$.

We now split ρ_n into the interface part $\hat{\rho}_n$ and the $O(h^2)$ part and treat them differently. The smoothness of $\hat{\rho}_n$ in time allows us to prove the following lemma.

Lemma 3. *Let $\hat{\varepsilon}_n = \sum_{j=1}^{n} (D^{-1}C)^{n-j} D^{-1} \hat{\rho}_{j-1}$ be the global error due to the interface. Then*

$$\|\hat{\varepsilon}_n\|_A \leq 4\tau^2 \cdot \max_{0 \leq l \leq n-1} \|X_2 A X_1 z_l\|_{A^{-1}} + O(\tau^4),$$

where $z_0 = -\mathscr{L}\partial_t u(\cdot, t_{1/2})$ and $z_l = -\mathscr{L}\partial_t^2 u(\cdot, t_l)$ for $l \geq 1$.

Since $\|u\|_{H^1(\Omega)}$ is spectrally equivalent to $\|u(\cdot)\|_A$, we can use Lemma 3 to obtain a bound for $\|\varepsilon_n\|_{H^1(\Omega)}$. To do so, we estimate

$$\|X_2 A X_1 z_l\|_{A^{-1}} = \|A^{-1/2}(I - X_1) A X_1 z_l\|_2 \leq \|A^{1/2} X_1 z_l\|_2 + \sqrt{\|X_1 A^{-1} X_1\|_2} \cdot \|A X_1 z_l\|_2.$$

But $X_1 A^{-1} X_1 = S_1^{-1}$, where S_1 is the Schur complement of A with respect to the interface. Thus, we can invoke the well-known Sobolev estimate [17, Lemma 4.11], cf. [2], which states that for a decomposition of Ω into shape-regular, conforming subdomains with diameter H, we have the condition number estimate

$$\kappa(S_1) := \|S_1\|_2 \|S_1^{-1}\|_2 \leq \frac{C}{Hh}.$$

Since A has been scaled in such a way that $\|S_1\|_2 = O(1)$, we conclude that $\|S_1^{-1}\|_2 \leq Ch^{-1}H^{-1}$. Additionally, since there are $O(h^{-1})$ points per interface and $O(H^{-1})$ interfaces, we have $\|X_1 z_l\|_2 = O(h^{-1/2}H^{-1/2})$. Combining these estimates leads to our main result.

Theorem 1. *Let Ω be partitioned into shape-regular, conforming subdomains Ω_i with diameter $\leq H$. Then for $\tau = \gamma h^\alpha$ for $\gamma > 0$ and $\alpha \geq 1$, the error of the CNPC method satisfies*

$$\|\varepsilon_n\|_{H^1(\Omega)} \leq \frac{Ch^\beta}{H}, \tag{4}$$

where $\beta = \min\{2\alpha - 1, 2\}$.

Thus, for a fixed number of subdomains, the method is second order if and only if $\alpha \geq 3/2$. For $\alpha = 1$, i.e., for $\tau = O(h)$, the method is only first order, unlike the classical CN method; this is due to the local inconsistency near subdomain interfaces.

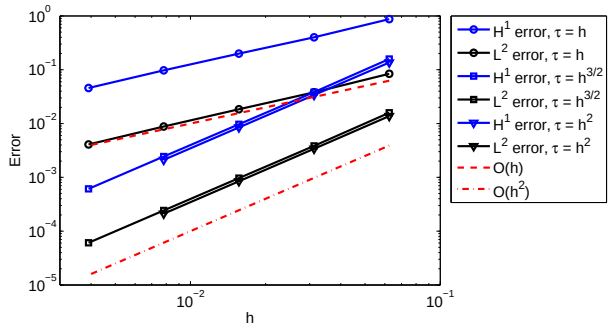


Fig. 4 Error of the CNPC scheme for the 2D heat equation $u_t - \Delta u = g(x,y,t)$ on $\Omega = (0,1)^2$.

4 Numerical results

We apply the CNPC method to solve

$$\partial_t u - \Delta u = g(x,y,t), \qquad (x,y) \in \Omega = (0,1) \times (0,1),$$

The domain Ω is decomposed into 4×4 equal subdomains, and the PDE discretized using a standard 5-point finite difference stencil in space. The initial conditions $u(x,y,0)$ and the source term $g(x,y,t)$ are chosen so that the exact solution is $u(x,y,t) = \sin(3\pi x)(1-e^{2y})(1-e^{y-1})\sqrt{1+t}$. Figure 4 shows the maximum L^2 and H^1 error of the method over the time interval $t \in (0,1)$, with $\tau = h^\alpha$ for $\alpha = 1, \frac{3}{2}, 2$. As predicted by Theorem 1, the error behaves like $O(h)$ for $\tau = h$, and $O(h^2)$ for $\alpha = \frac{3}{2}$ and 2. Moreover, we also see that using the finer time step $\tau = h^2$ only improves the error marginally when compared to $\tau = h^{3/2}$.

Table 1 shows the error of the method for $\tau = h$ and $\tau = h^{3/2}$, when Ω is decomposed into $N \times N$ subdomains with $N = 1/H$. We see that for $\tau = h$, the estimate (4) is sharp; indeed, the errors are approximately constant along the diagonals, except for the column $N = 2$. For $\tau = h^{3/2}$, the estimate is too conservative, as the error does not deteriorate as the number of subdomains increases. This appears to be a 2D effect, since the estimate is sharp for $\tau = h^{3/2}$ in the 1D case. Thus, there appears to be a subtle interplay between temporal and spatial interpolation errors that gives rise to this "superconvergence" behavior.

Conclusions and outlook. The CNPC method allows one to solve diffusion problems in parallel to second-order accuracy without iterating, provided $\tau = O(h^{3/2})$ or smaller. For 3D problems, the Schur complement (3) becomes much denser; one alternative is to use a two-level approach, by first correcting the face values using explicit edge and vertex values, and then correct the edge and vertex values using the face values. The error analysis for this variant, as well as for more general equations (e.g. the advection-diffusion equation), will be the subject of a future paper.

Table 1 Maximum L^2 error for the 2D example .

	$\tau = h$				$\tau = h^{3/2}$			
	Subdomains per direction ($N = 1/H$)				Subdomains per direction ($N = 1/H$)			
$n = 1/h$	2	4	8	16	2	4	8	16
16	7.540e-02	2.347e-01	3.300e-01		5.888e-02	6.585e-02	7.165e-02	
32	2.265e-02	1.399e-01	2.330e-01	3.185e-01	1.448e-02	1.397e-02	1.392e-02	1.402e-02
64	1.291e-02	7.602e-02	1.382e-01	2.391e-01	3.607e-03	3.425e-03	3.296e-03	3.168e-03
128	6.941e-03	4.006e-02	7.405e-02	1.397e-01	9.010e-04	8.513e-04	8.107e-04	7.571e-04
256	3.597e-03	2.053e-02	3.838e-02	7.426e-02	2.252e-04	2.124e-04	2.015e-04	1.860e-04

References

1. Bennequin, D., Gander, M., Halpern, L.: A homographic best approximation problem with application to optimized Schwarz waveform relaxation. Math. Comp. **78**(265), 185–223 (2009)
2. Brenner, S.C.: The condition number of the Schur complement in domain decomposition. Numer. Math. **83**, 187–203 (1999)
3. Cai, X.C.: Additive Schwarz algorithms for parabolic convection-diffusion equations. Numer. Math. **60**, 41–61 (1991)
4. Dawson, C.N., Du, Q., Dupont, T.F.: A finite difference domain decomposition algorithm for numerical solution of the heat equation. Math. Comp. **57**, 63–71 (1991)
5. Farhat, C., Lesionne, M., Le Tallec, P., Pierson, K., Rixen., D.: FETI-DP: a dual-primal unified FETI method — part I: a faster alternative to the two-level FETI method. Internat. J. Numer. Methods Engrg. **50**, 1523–1544 (2001)
6. Gander, M.J., Hairer, E.: Nonlinear convergence analysis for the parareal algorithm. In: Domain Decomposition Methods in Science and Engineering XVII, pp. 45–56. Springer (2008).
7. Gander, M.J., Halpern, L.: Optimized Schwarz waveform relaxation for advection reaction diffusion problems. SIAM J. Numer. Anal. **45**, 666–697 (2007)
8. Gander, M.J., Stuart, A.: Space-time continuous analysis of waveform relaxation for the heat equation. SIAM J. Sci. Comput. **19**(6), 2014–2031 (1998)
9. Giladi, E., Keller, H.B.: Space-time domain decomposition for parabolic problems. Numer. Math. **93**, 279–313 (2002)
10. Janssen, J., Vandewalle, S.: Multigrid waveform relaxation on spatial finite element meshes: the continuous case. SIAM J. Sci. Comput. **17**, 133–155 (1996)
11. Kuznetsov, Y.A.: Overlapping domain decomposition methods for parabolic problems. In: Domain Decomposition Methods in Science and Engineering. AMS (1992)
12. Kwok, F.: A parallel predictor-corrector crank–nicolson method for the solution of parabolic equations. submitted (2012)
13. Lions, J.L., Maday, Y., Turinici, G.: A parareal in time discretization of PDE's. C.R. Acad. Sci. Paris, Série I **332**, 661–668 (2001)
14. Meurant, G.: Numerical experiments with domain decomposition methods for parabolic problems on parallel computers. In: Domain Decomposition Methods for Partial Differential Equations. SIAM (1991)
15. Rempe, M.J., Chopp, D.L.: A predictor-corrector algorithm for reaction-diffusion equations associated with neural activity on branched structures. SIAM J. Sci. Comput. **28**, 2139–2161 (2006)
16. Shi, H.S., Liao, H.L.: Unconditional stability of corrected explicit-implicit domain decomposition algorithms for parallel approximation of heat equations. SIAM J. Numer. Anal. **44**(4), 1584–1611 (2006)
17. Toselli, A., Widlund, O.B.: Domain Decomposition Methods — Algorithms and Theory, *Springer Series in Computational Mathematics*, vol. 34. Springer, Berlin Heidelberg (2005)

Heterogeneous coupling for implicitly described domains

Christian Engwer[1] and Sebastian Westerheide[1]

1 Introduction

Many applications in physics, biology or chemistry exhibit complex geometrical shapes. Often these models feature partial differential equations (PDEs) on the complex shaped domain and its surface. At the same time the domain might be time-dependent, e.g. in cell biology the shape of a cell depends on its internal state and couples back to the cell metabolism, cf. [12]. Modern imaging techniques yield high resolution data of microscopic structures and thus allow us to exploit direct simulations.

Constructing suitable meshes for complex geometries is a very involved task, thus methods to decouple the computational mesh from the geometry are of great interest. In the context of Fictitious Domain Methods a wide range of methods was developed; we want to mention explicitly the Unfitted Finite Element Method [2, 13, 4], which we build upon. These methods formulate the original problem as a problem embedded in a larger domain. Different ways of incorporating the, now internal, boundary conditions are described in the literature. Examples for applications to coupled problems can be found using XFEM [9, 10], or using fictitious domain and mortar methods [1]. Many of these methods have been developed for engineering applications and are not directly applicable to biological problems as certain processes, e.g. topology changes can not be captured. An alternative class of methods uses implicit domain descriptions as level sets [17], or phase-field models [5, review paper]. Both approaches have been applied to coupled problems (e.g. [6, 18]), but due to the diffusive representation of the coupling interface these methods can lead to numerical artifacts, including spurious fluxes.

In this work we present a new approach to incorporate processes on manifolds in a heterogeneous domain-decomposition framework for implicitly described geometries. Although using a level set formulation, we avoid a diffuse coupling interface by utilizing an explicit reconstruction. It uses concepts of the Unfitted Finite Element Method and can be directly applied to image data.

Outline. The paper is structured as follows. In section 2 we discuss how domains can be described implicitly and in the following section we introduce the model problem. Section 4 describes the numerical scheme, starting with the Unfitted Discontinuous Galerkin approach for volume equations and then presenting a consistent approach for equations on the surface as well as the way of imposing coupling conditions. Finally, a numerical example is discussed in section 5.

[1] Institute for Computational and Applied Mathematics, University of Muenster, Germany, e-mail: {christian.engwer}{sebastian.westerheide}@uni-muenster.de

2 Implicitly described domains

For each $t \in [0,T]$, $T > 0$, let $\Omega(t) \subset \mathbb{R}^n$ be a Lipschitz bounded domain and $\Gamma(t)$ its boundary, with ν denoting the outward pointing unit normal vector field to $\Gamma(t)$.

By embedding $\Omega(t)$ in a larger stationary domain $\hat{\Omega}$, it is possible to describe $\Omega(t)$ using the so-called level set approach [14]. It captures the geometric information and motion of a moving interface from an Eulerian point of view in terms of a level set function and an associated PDE. A level set function is a scalar function $\Phi(x,t)$ defined in $\hat{\Omega} \times [0,T]$ with

$$\Phi(x,t) \begin{cases} < 0 & \text{for } x \in \Omega(t), \\ = 0 & \text{for } x \in \Gamma(t), \\ > 0 & \text{else}, \end{cases}$$

like illustrated in Figure 1. For each t the interface $\Gamma(t)$ corresponds to the zero level set $\Phi^{-1}(0) := \{x \in \hat{\Omega} \mid \Phi(x,t) = 0\}$. $\Phi(x,t)$ satisfies the level set advection equation

$$\Phi_t + \mathbf{v} \cdot \nabla \Phi = 0,$$

where $\mathbf{v}(x,t)$ is a velocity field corresponding to the evolution of $\Omega(t)$ and $\Gamma(t)$.

The level set approach allows for an elegant treatment of complex geometrical morphologies with potential topology changes in a fully implicit way, as discrete versions of Φ can be defined using a fixed grid on $\hat{\Omega}$. It is convenient to choose an appropriate $\hat{\Omega}$ which allows to use a simple Cartesian grid.

In this paper we only consider static domains, i.e. $\mathbf{v} \equiv 0$. Eulerian formulations of PDEs on moving domains contain additional terms corresponding to the transport of information induced by domain movement, the so-called material derivatives. The numerical schemes we present in section 4 are extended accordingly by appropriate transport terms.

3 Model problem

Let u_1 and u_2 denote the concentrations of two scalar quantities on a static domain Ω and its surface Γ, respectively. Conservation of these quantities with a diffusive flux $-\mathscr{D}_1 \nabla u_1$ in Ω and a diffusive surface flux $-\mathscr{D}_2 \nabla_\Gamma u_2$ together with an additional reactive process on Γ leads to the model problem we want to consider. Given some initial values $u_1(\cdot,0)$ and $u_2(\cdot,0)$, it reads

$$\partial_t u_1 = \nabla \cdot (\mathscr{D}_1 \nabla u_1) \qquad \text{in} \quad \Omega \times (0,T], \tag{1a}$$

$$\partial_t u_2 = \nabla_\Gamma \cdot (\mathscr{D}_2 \nabla_\Gamma u_2) + r_2\big(u_1|_\Gamma, u_2\big) \qquad \text{on} \quad \Gamma \times (0,T], \tag{1b}$$

$$\mathscr{D}_1 \nabla u_1 \cdot \nu = r_1\big(u_1|_\Gamma, u_2\big) \qquad \text{on} \quad \Gamma \times (0,T]. \tag{1c}$$

Here, ∇_Γ denotes the tangential surface gradient as well as the induced surface divergence. $\mathscr{D}_1$ and $\mathscr{D}_2$ are the particular volume and surface diffusivity tensors, i.e. $\mathscr{D}_2$ maps the tangent space of Γ into itself at every point. r_1 together with r_2 are potentially nonlinear terms which couple the processes in Ω and Γ. For example, they could describe transitions between u_1 and u_2. The coupling in equation (1a) is due to its Robin-like boundary condition (1c), whereas r_2 appears as a standard surface reaction term in equation (1b).

4 Heterogeneous coupling

We propose a new numerical scheme for solving problems like model problem (1). It is based on the Unfitted Discontinuous Galerkin method (UDG) for solving PDEs in Ω and a level set based extension to surface PDEs. The method of lines [16] is used to split spatial and temporal operators. A semi-discretization in space yields: Find $(u_{1,h}, u_{2,h}) \in L^2(0,T;V_{1,h}) \times L^2(0,T;V_{2,h})$ such that for each $t \in (0,T]$

$$\begin{aligned} t_{\text{vol}}(u_{1,h}, v_{1,h}, t) + a_{\text{vol}}(u_{1,h}, v_{1,h}, t) + c_1(u_{1,h}, u_{2,h}, v_{1,h}, t) = 0 \quad \forall v_{1,h} \in V_{1,h}, \\ t_{\text{sur}}(u_{2,h}, v_{2,h}, t) + a_{\text{sur}}(u_{2,h}, v_{2,h}, t) + c_2(u_{1,h}, u_{2,h}, v_{2,h}, t) = 0 \quad \forall v_{2,h} \in V_{2,h}, \end{aligned} \tag{2}$$

where $V_{1,h}$ and $V_{2,h}$ denote discrete function spaces. The operators t_{vol} and t_{sur} correspond to the two time derivatives $\partial_t u_1$ and $\partial_t u_2$ in problem (1). The elliptic diffusion terms of equations (1a) and (1b) are contained in the operators a_{vol} and a_{sur}, respectively, and c_1 and c_2 are coupling operators which correspond to the terms r_1 and r_2. To get a fully discrete scheme, different time discretization schemes can be used.

Bulk discretization: The Unfitted Discontinuous Galerkin method. To treat the bulk equations (1a, 1c), we consider the UDG method [4], which is a general approach for simulations on complicated domains. It uses the concepts of the Unfitted Finite Element Method [2, 13] and discretizes PDEs on an unfitted mesh, i.e. the domain boundary Γ is not resolved by the mesh. For an easy implementation, this so called *fundamental mesh* is chosen to be the same mesh as for the discrete level set function. Shape functions are defined on the unfitted mesh and their support is restricted to Ω. We use a Discontinuous Galerkin (DG) discretization. This allows to easily incorporate local mass conservation and to use higher order shape functions.

Based on the fundamental mesh $\mathscr{T}(\hat{\Omega}) := \{\hat{E}_0, \ldots, \hat{E}_{M-1}\}$, a Finite Element mesh for domain Ω is defined by intersecting Ω and $\mathscr{T}(\hat{\Omega})$ (see Figure 1):

$$\mathscr{T}(\Omega) := \{E_n = \Omega \cap \hat{E}_n \mid \hat{E}_n \in \mathscr{T}(\hat{\Omega}),\ |E_n| > 0\}.$$

The elements E_n can be arbitrarily shaped and in general will not be convex. Using standard DG shape functions on $\mathscr{T}(\hat{\Omega})$ with their support restricted to the elements in $\mathscr{T}(\Omega)$, the resulting Finite Element space is defined by

$$V_{1,h} := \left\{ v \in L^2(\Omega) \;\middle|\; v|_{E_n} \in P_k \ \forall E_n \in \mathscr{T}(\Omega) \right\},$$

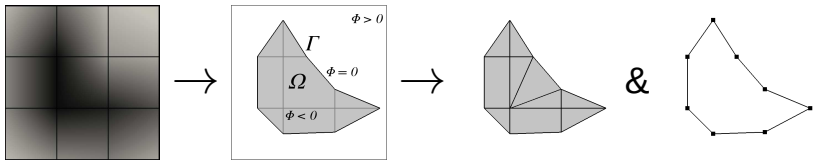


Fig. 1 Given the fundamental mesh $\mathscr{T}(\hat{\Omega})$ and a piecewise linear level set function Φ (left), the domain Ω and the Finite Element mesh $\mathscr{T}(\Omega)$ are defined. Local triangulations of its cells E_n and ∂E_n yield a partition of Ω into integration parts $\{E_{n,k}\}$ and a piecewise linear reconstruction of Γ.

P_k being the space of polynomial functions of degree k. $V_{1,h}$ is discontinuous on the internal skeleton $\Gamma_{\text{int}} := \{\gamma_{n,m} = \partial E_n \cap \partial E_m \mid E_n, E_m \in \mathscr{T}(\Omega),\ E_n \neq E_m,\ |\gamma_{n,m}| > 0\}$, with $|\gamma_{n,m}|$ denoting the codimension one volume of $\gamma_{n,m}$, but not on the external skeleton $\Gamma_{\text{ext}} := \{\gamma_n = \partial E_n \cap \partial\Omega \mid E_n \in \mathscr{T}(\Omega),\ |\gamma_n| > 0\}$. To each $\gamma_{n,m} = \gamma_{m,n}$, we assign unit normal vector fields $\mathbf{n}_{E_n} \equiv -\mathbf{n}_{E_m}$ and arbitrarily choose $\mathbf{n} := \mathbf{n}_{E_n}$. Using the DG formulation described in [15], the operators which result from eq. (1a) read:

$$t_{\text{vol}}(u_{1,h}, v_{1,h}, t) := \frac{d}{dt} \sum_{E_n \in \mathscr{T}(\Omega)} \int_{E_n} u_{1,h}\, v_{1,h}\, dV,$$

$$a_{\text{vol}}(u_{1,h}, v_{1,h}, t) := \sum_{\gamma_{n,m} \in \Gamma_{\text{int}}} \int_{\gamma_{n,m}} \varepsilon \langle (\mathscr{D}_1 \nabla v_{1,h}) \cdot \mathbf{n} \rangle [u_{1,h}] - \langle (\mathscr{D}_1 \nabla u_{1,h}) \cdot \mathbf{n} \rangle [v_{1,h}]\, ds$$
$$+ \sum_{E_n \in \mathscr{T}(\Omega)} \int_{E_n} (\mathscr{D}_1 \nabla u_{1,h}) \cdot \nabla v_{1,h}\, dV + \sum_{\gamma_{n,m} \in \Gamma_{\text{int}}} \frac{\sigma}{|\gamma_{n,m}|^\beta} \int_{\gamma_{n,m}} [u_{1,h}][v_{1,h}]\, ds.$$

Here, σ and β are appropriate stabilization parameters and $\varepsilon = \pm 1$. Furthermore, $[\cdot]$ denotes the jump of a function $v \in V_{1,h}$ on the interface between two adjacent elements E_n, E_m which is defined as $[v] := v|_{\partial E_n} - v|_{\partial E_m}$ and the average $\langle \cdot \rangle$ is defined as $\langle v \rangle := \frac{1}{2}\left(v|_{\partial E_n} + v|_{\partial E_m}\right)$.

Assembling the local stiffness matrix requires integration over the volume of each element E_n and different parts of its surface ∂E_n. As these mesh elements might exhibit very complicated shapes, quadrature rules based on interpolation functions are not directly applicable. Integation on the fundamental mesh also does not work, since shape functions are discontinuous. In order to guarantee accurate evaluation of integals in an efficient manner, quadrature rules for irregular shaped elements are constructed using a local triangulation of E_n. To do so, E_n is subdivided into a disjoint set $\{E_{n,k}\}_k$ of simple geometric objects, i.e. simplices and hypercubes. For each of these *integration parts* an efficient Gauss type quadrature rule is available. For a piecewise linear approximation of the level set function, the local triangulation can be efficiently constructed by applying a modified marching cubes algorithm [4].

Extension to surface equations. The pure surface part of model problem (1) without the coupling term r_2 reads

$$\partial_t u_2 = \nabla_\Gamma \cdot (\mathscr{D}_2 \nabla_\Gamma u_2) \quad \text{on} \quad \Gamma \times (0, T]. \tag{3}$$

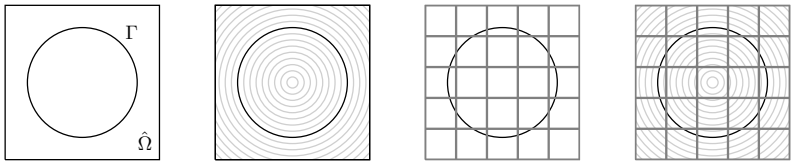


Fig. 2 From left to right: Surface Γ embedded into the larger level set domain $\hat{\Omega}$, Γ and some other level sets Γ_r of Φ, the same together with a Cartesian grid on $\hat{\Omega}$.

To treat this equation, we combine the DG method with an implicit surface Finite Element approach which was introduced in [7]. Similar to the method described in [7], we make use of the implicit level set description of Γ. The basic idea is to extend a surface diffusion equation like (3) and its solution to the whole level set domain $\hat{\Omega}$ by simultaneously formulating the $(n-1)$-dimensional PDE on all level surfaces of Φ. The resulting n-dimensional problem is solved using a DG discretization on an arbitrary triangulation of $\hat{\Omega}$. See also Figure 2. The solution of the original surface problem is then obtained by restricting the higher dimensional solution to Γ.

In particular, we use that we can partition $\hat{\Omega}$ into level surfaces

$$\Gamma_r := \left\{ x \in \hat{\Omega} \;\middle|\; \Phi(x) = r \right\}$$

with $\bigcup_{r\in(\Phi_{\min},\Phi_{\max})} \Gamma_r = \hat{\Omega}$, $\Phi_{\min} := \inf_{x\in\hat{\Omega}} \Phi(x)$, $\Phi_{\max} := \sup_{x\in\hat{\Omega}} \Phi(x)$. Note that $\Gamma = \Gamma_0$. First, we create a suitable extension $\mathscr{D}_2^{\Phi}$ of the surface diffusivity tensor $\mathscr{D}_2$ to the level set domain $\hat{\Omega}$, such that we do not have any diffusion normal to any level surface. In detail, $\mathscr{D}_2^{\Phi}$ is chosen such that $\mathscr{D}_2^{\Phi}\big|_{\Gamma} = \mathscr{D}_2$ and

$$\mathscr{D}_2^{\Phi} \nu^{\perp} \cdot \nu = 0 \quad \text{in} \quad \hat{\Omega} \times (0,T] \tag{4}$$

for every tangential vector $\nu^{\perp}$, where we now denote by ν the outward pointing unit normal vector field to every level surface. Then the elliptic surface differential operator ∇_{Γ} is extended to each level surface Γ_r yielding a differential operator ∇_{Φ}. Using these extensions, (3) is formulated on all level surfaces Γ_r. This results in the n-dimensional equation

$$\partial_t u_2 = \nabla_{\Phi} \cdot \left(\mathscr{D}_2^{\Phi} \nabla_{\Phi} u_2 \right) \quad \text{in} \quad \hat{\Omega} \times (0,T].$$

Assuming that the level set function Φ is differentiable and satisfies a non-degeneracy condition $\nabla\Phi \neq 0$ in $\hat{\Omega} \cup \partial\hat{\Omega}$, we can follow the approach from [7, Remark 3.3] and reformulate the extended tangential surface divergence operator ∇_{Φ}. This results in an equivalent equation

$$\partial_t u_2 |\nabla\Phi| = \nabla \cdot \left(\tilde{\mathscr{D}}_2^{\Phi} \nabla u_2 \right) \quad \text{in} \quad \hat{\Omega} \times (0,T], \tag{5}$$

with a modified diffusion tensor $\tilde{\mathscr{D}}_2^{\Phi} := |\nabla\Phi| \mathscr{D}_2^{\Phi} \mathscr{P}_{\Phi}$. At every point in $\hat{\Omega}$, $\mathscr{P}_{\Phi}$ is the operator which projects onto the tangent space of the corresponding level sur-

face. Equation (5) is a usual parabolic diffusion equation in $\mathbb{R}^n$ with a special mass density. In order to define a well-posed problem it has to be supplemented by initial values and an appropriate boundary condition for u_2 on $\partial\hat{\Omega}$. We choose initial values which are an arbitrary but continuous extension of the original initial values chosen for equation (1b) and use the natural no-flux boundary condition $\tilde{\mathscr{D}}_2^{\Phi}\nabla u_2 \cdot \nu_{\partial\hat{\Omega}} = 0$, with the outer unit normal $\nu_{\partial\hat{\Omega}}$. Note that the restricted solution on a particular level surface Γ_r only depends on the values of data on that surface as we do not have any diffusion in the normal direction due to equation (4). Therefore it is independent of the solutions on any other level surface. It can, however, be related to these solutions by the extensions of the data. Furthermore, it is not affected by the artificial boundary condition as long as Γ_r does not intersect $\partial\hat{\Omega}$. Further note that the solution on Γ, i.e. $u_2|_\Gamma$, solves equation (3).

The initial-boundary-value problem resulting from equation (5) can be discretized on the fundamental mesh $\mathscr{T}(\hat{\Omega})$ by usual grid-based numerical methods. Using the same DG formulation as for the volume part, we obtain:

$$t_{\mathrm{sur}}(u_{2,h}, v_{2,h}, t) := \frac{d}{dt} \sum_{\hat{E}_n \in \mathscr{T}(\hat{\Omega})} \int_{\hat{E}_n} u_{2,h}\, v_{2,h}\, |\nabla \Phi|\, dV,$$

$$\begin{aligned} a_{\mathrm{sur}}(u_{2,h}, v_{2,h}, t) := &\sum_{\hat{\gamma}_{n,m} \in \hat{\Gamma}_{\mathrm{int}}} \int_{\hat{\gamma}_{n,m}} \varepsilon \left\langle \left(\tilde{\mathscr{D}}_2^{\Phi}\nabla v_{2,h}\right)\cdot \mathbf{n} \right\rangle [u_{2,h}] - \left\langle \left(\tilde{\mathscr{D}}_2^{\Phi}\nabla u_{2,h}\right)\cdot \mathbf{n} \right\rangle [v_{2,h}]\, ds \\ &+ \sum_{\hat{E}_n \in \mathscr{T}(\hat{\Omega})} \int_{\hat{E}_n} \left(\tilde{\mathscr{D}}_2^{\Phi}\nabla u_{2,h}\right)\cdot \nabla v_{2,h}\, dV + \sum_{\hat{\gamma}_{n,m} \in \hat{\Gamma}_{\mathrm{int}}} \frac{\sigma}{|\hat{\gamma}_{n,m}|^{\beta}} \int_{\hat{\gamma}_{n,m}} [u_{2,h}][v_{2,h}]\, ds. \end{aligned}$$

Here, we choose the discrete function space

$$V_{2,h} := \left\{ v \in L^2(\hat{\Omega}) \;\middle|\; v|_{\hat{E}_n} \in P_k \;\forall \hat{E}_n \in \mathscr{T}(\hat{\Omega}) \right\},$$

and the jump $[\cdot]$ and average $\langle \cdot \rangle$ act on functions from $V_{2,h}$, targeting discontinuities that lie on the internal skeleton of $\mathscr{T}(\hat{\Omega})$, which is defined by

$$\hat{\Gamma}_{\mathrm{int}} := \left\{ \hat{\gamma}_{n,m} = \partial\hat{E}_n \cap \partial\hat{E}_m \;\middle|\; \hat{E}_n, \hat{E}_m \in \mathscr{T}(\hat{\Omega}),\ \hat{E}_n \neq \hat{E}_m,\ |\hat{\gamma}_{n,m}| > 0 \right\}.$$

Explicit coupling of bulk and surface. The volume coupling operator c_1 results from the way DG formulations include boundary conditions of Robin type. For boundary condition (1c) we get

$$c_1(u_{1,h}, u_{2,h}, v_{1,h}, t) := -\sum_{\gamma_n \in \Gamma_{\mathrm{ext}}} \int_{\gamma_n} r_1\left(u_{1,h}|_\Gamma, u_{2,h}|_\Gamma\right) v_{1,h}|_\Gamma\, ds.$$

The surface coupling operator c_2 is imposed directly along Γ by choosing

$$c_2(u_{1,h}, u_{2,h}, v_{2,h}, t) := -\sum_{\gamma_n \in \Gamma_{\mathrm{ext}}} \int_{\gamma_n} r_2\left(u_{1,h}|_\Gamma, u_{2,h}|_\Gamma\right) v_{2,h}|_\Gamma\, ds,$$

such that the native surface reaction term r_2 from equation (1b) now acts like the lower order term in a Robin-like inner boundary condition. The integrals over each γ_n are efficiently evaluated using the local triangulation of the bulk discretization. In each time step, this results in a globally coupled block system $A = \begin{pmatrix} A_{vol} & C_1 \\ C_2 & A_{sur} \end{pmatrix}$, which can be solved fully coupled or using a Schwarz type iteration.

5 Numerical example and conclusion

We compute a problem from cell biology. Prior to cell division, the shape of a single yeast cell can be idealized as a circular domain $\Omega \subset \mathbb{R}^2$ whose surface Γ is the cell membrane. We use eq. (1) to model the intracellular pathway of a protein known as CDC42, where u_1 and u_2 denote its bulk and surface concentration. Diffusion driven instabilities lead to clustering of CDC42 on the membrane, which triggers the sprouting of a bud in areas of high concentration. The model uses coupling terms $r_2(u_1,u_2) := -r_1(u_1,u_2) := k_1 \cdot u_1 u_2^2 + k_2 \cdot u_1 u_2 - k_3 \cdot u_2$, $k_1 := 0.0036$, $k_2 := 0.0067$, $k_3 := 0.01733$, which describe transitions between CDC42 inside of the cell and on its membrane, and constant diffusivities $\mathscr{D}_1 := 10$, $\mathscr{D}_2 := 0.0025 =: \mathscr{D}_2^{\Phi}$.

In our simulation, we use a level set domain $\hat{\Omega} = [0,1]^2$ and a Cartesian fundamental mesh $\mathscr{T}(\hat{\Omega})$ which contains 32×32 elements. The cell Ω is positioned in the center of $\hat{\Omega}$. It is described by a level set function $\Phi(x) := \|x - (0.5, 0.5)^T\| - 0.35$ which is approximated using Q1 Finite Elements on $\mathscr{T}(\hat{\Omega})$.

The discretization is done using polynomial degree $k = 1$. For bulk discretization we choose $\varepsilon = -1$, the Interior Penalty Galerkin scheme. For the surface discretization we use $\varepsilon = +1$, the Nonsymmetric Interior Penalty Galerkin scheme. The resulting semi-discretized problem (2) is solved using Newton's method for linearization and the fractional step θ-method [11] for time discretization. As shown in Figure 3, random generated initial values for u_1 and u_2 lead to the expected localization of u_2 on the membrane.

Conclusion. The proposed approach yields a unified setting for coupled volume and surface problems. The same infrastructure can be used to implement the discretization of both the volume and the surface part. Coupling conditions are handled explicitly along the surface in an efficient way without additional effort. At the same time we use an implicit description of the domain which makes the method completely independent of the problem's geometry. This level set based Eulerian formulation makes the approach a promising tool for biological problems, especially those which involve strongly evolving domains with potential topology changes.

Future topics may include the application to evolving domain problems or a thorough error analysis.

Acknowledgements The authors thank Wolfgang Giese (HU Berlin) for providing the budding yeast model which is based on [12]. All implementations were done using the frameworks DUNE [3] and DUNE-UDG [8].

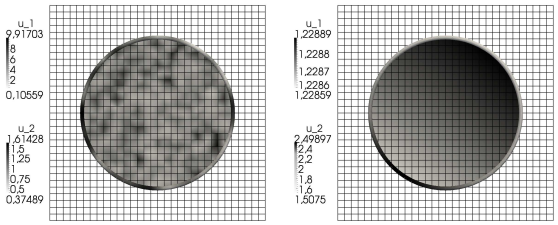


Fig. 3 Left: Initial values on a circular shaped domain Ω and its surface Γ. Right: Simulation result at final time $T = 500$, using polynomial degree $k = 1$ and time step $dt = 0.5$; note the localization of u_2 on Γ at the lower left.

References

1. Baaijens, F.: A fictitious domain/mortar element method for fluid-structure interaction. International Journal for Numerical Methods in Fluids **35**(7), 743–761 (2001)
2. Barrett, J.W., Elliott, C.M.: Fitted and unfitted finite-element methods for elliptic equations with smooth interfaces. IMA J. Numer. Anal. **7**(3), 283–300 (1987)
3. Bastian, P., Blatt, M., Dedner, A., Engwer, C., Klfkorn, R., Kornhuber, R., Ohlberger, M., Sander, O.: A generic grid interface for parallel and adaptive scientific computing. part ii: implementation and tests in dune. Computing **82**, 121–138 (2008)
4. Bastian, P., Engwer, C.: An unfitted finite element method using discontinuous galerkin. International Journal for Numerical Methods in Engineering **79**(12), 1557–1576 (2009)
5. Boettinger, W., Warren, J.: Phase-field simulation of solidification. Annual Review of Materials Research (2002)
6. Cirak, F., Radovitzky, R.: A lagrangian–eulerian shell–fluid coupling algorithm based on level sets. Computers & Structures **83**(6), 491–498 (2005)
7. Dziuk, G., Elliott, C.: Eulerian finite element method for parabolic pdes on implicit surfaces. Interfaces and Free Boundaries **10**, 119–138 (2008)
8. Engwer, C., Heimann, F.: Dune-udg: A cut-cell framework for unfitted discontinuous galerkin methods. In: Advances in DUNE, pp. 89–100. Springer (2012)
9. Farsad, M., Vernerey, F., Park, H.: An extended finite element/level set method to study surface effects on the mechanical behavior and properties of nanomaterials. International Journal for Numerical Methods in Engineering **84**(12), 1466–1489 (2010)
10. Gerstenberger, A., Wall, W.: An extended finite element method/lagrange multiplier based approach for fluid–structure interaction. Computer Methods in Applied Mechanics and Engineering **197**(19), 1699–1714 (2008)
11. Glowinski, R.: Viscous flow simulation by finite element methods and related numerical techniques. In: Progress and supercomputing in computational fluid dynamics, pp. 173–210 (1985)
12. Goryachev, A., Pokhilko, A.: Dynamics of cdc42 network embodies a turing-type mechanism of yeast cell polarity. FEBS Letters **582**(10), 1437–1443 (2008)
13. Hansbo, A., Hansbo, P.: An unfitted finite element method, based on nitsches method, for elliptic interface problems. Computer Methods in Applied Mechanics and Engineering **191**(47-48), 5537–5552 (2002)
14. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**(1), 12–49 (1988)
15. Rivière, B., Bastian, P.: Discontinuous galerkin methods for two-phase flow in porous media. Tech. Rep. 2004–28, IWR (SFB 359), University of Heidelberg (2004)
16. Schiesser, W.: The numerical method of lines. Academic Press Inc. (1991). Integration of partial differential equations
17. Sethian, J., Strain, J.: Crystal growth and dendritic solidification. J. Comput. Phys. **98**(2), 231–253 (1992)
18. Teigen, K., Li, X., Lowengrub, J., Wang, F., Voigt, A.: A diffuse-interface approach for modelling transport, diffusion and adsorption/desorption of material quantities on a deformable interface. Communications in Mathematical Sciences **7**(4), 1009–1037 (2009)

NKS Method for the Implicit Solution of a Coupled Allen-Cahn/Cahn-Hilliard System*

Chao Yang[1], Xiao-Chuan Cai[2], David E. Keyes[3], and Michael Pernice[4]

1 Coupled Allen-Cahn/Cahn-Hilliard system

Coupled Allen-Cahn/Cahn-Hilliard (AC/CH) systems, often found in phase-field simulations, are prototype systems that admit simultaneous ordering and phase separation. Numerical methods to solve coupled AC/CH systems are studied in e.g., [2, 6, 8, 9, 10, 11]. However, except for [9] and [10], the above works are based on explicit methods that require very small time step size to advance the solution and need many time steps for long time integrations. Fully implicit methods enjoy an advantage that the stability limit on the time step size is greatly relaxed. The purpose of this paper is to study efficient and scalable algorithms based on domain decomposition methods for the fully implicit solution of a coupled AC/CH system.

There are several different ways to couple the AC and the CH equations. Among them we restrict our study to the original form introduced in [3], which is

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla \cdot c(u,v) \nabla \dfrac{\delta E(u,v)}{\delta u}, \\ \dfrac{\partial v}{\partial t} = -\dfrac{c(u,v)}{\rho} \dfrac{\delta E(u,v)}{\delta v}. \end{cases} \tag{1}$$

where u and v are functions of $\mathbf{x} \in \Omega \subset \mathbf{R}^2$ and $t \in [0, +\infty)$. Both u and v are bounded with restrictions: $u \in [0,1]$, $v \in [-1/2, 1/2]$ and $(u \pm v) \in [0,1]$. Here the first equation in (1) is the Cahn-Hilliard equation in which u represents a conserved concentration field for the phase separation; the second equation in (1) is the Allen-Cahn equation where v denotes a non-conserved order parameter for the anti-phase coarsening.

In (1), the mobility $c(u,v) = u(1-u)(1/4 - v^2)$ is degenerate at pure phases and the density ρ is a positive constant. The free energy functional $E(u,v)$ reads

[1] Chao Yang, Institute of Software, Chinese Academy of Sciences, Beijing 100190, and State Key Laboratory of High Performance Computing, Changsha 410073, China, e-mail: yangchao@iscas.ac.cn ·[2] Xiao-Chuan Cai, Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA, e-mail: cai@cs.colorado.edu ·[3] David E. Keyes, CEMSE Division, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia, e-mail: david.keyes@kaust.edu.sa ·[4] Michael Pernice, Idaho National Laboratory, Idaho Falls, ID 83415, USA e-mail: michael.pernice@inl.gov

* This work was supported in part by DE-FC02-06ER25784. The first author also received supports from NSFC under 61170075, 91130023 and 61120106005, and from 973 Program of China under 2011CB309701.

$$E(u,v)=\int_\Omega \left\{ \frac{\gamma}{2}(|\nabla u|^2+|\nabla v|^2)+\theta\left(\Phi(u+v)+\Phi(u-v)\right)+\frac{\alpha}{2}u(1-u)-\frac{\beta}{2}v^2 \right\} d\mathbf{x}, \quad (2)$$

where $\Phi(z)=z\ln z+(1-z)\ln(1-z)$ and γ, θ, α, β are all positive constants. It then follows that

$$\begin{aligned} \frac{\delta E}{\delta u} &= -\gamma\Delta u+\theta\Phi'(u+v)+\theta\Phi'(u-v)-\alpha(u-1/2), \\ \frac{\delta E}{\delta v} &= -\gamma\Delta v+\theta\Phi'(u+v)-\theta\Phi'(u-v)-\beta v. \end{aligned} \quad (3)$$

In the current study we consider periodic boundary conditions for both u and v. Other boundary conditions lead to similar numerical results and the performance of our proposed solver is not sensitive to them. The AC/CH system (1) is closed with the above boundary conditions and an initial condition $u=u^0$, $v=v^0$ at $t=0$.

2 Discretizations

We restrict our study in this paper to the case of a 2-dimensional square domain Ω. A second-order accurate cell-centered finite difference (CCFD) scheme on a uniform mesh is applied to the system. The details of the CCFD scheme is omitted here due to the page limit.

Special attention should be paid when considering the time integration of the AC/CH system (1). Because of the high-order spatial differentiation in the system, explicit methods become impractical due to the severe restriction on the time step size. In order to relax the restriction and obtain the steady-state solution in an efficient way, we use the fully implicit backward Euler scheme. We remark that due to the co-existence of both diffusive and anti-diffusive terms in the AC/CH system, the backward Euler scheme is not unconditionally stable. Other more efficient and accurate implicit schemes will be studied in a forthcoming paper.

After spatially discretizing the AC/CH system, u and v are replaced with their cell-centered values U and V respectively. Denote the spatial discretizations of the right-hand-sides in the two equations in (1) as $M(U,V)$ and $N(U,V)$ respectively, the nonlinear algebraic system arising at each time step reads

$$\begin{cases} \mathscr{M}_k(U^{k+1},V^{k+1}) := \dfrac{U^{k+1}-U^k}{\Delta t^k}-M(U^{k+1},V^{k+1})=0, \\ \mathscr{N}_k(U^{k+1},V^{k+1}) := \dfrac{V^{k+1}-V^k}{\Delta t^k}-N(U^{k+1},V^{k+1})=0, \end{cases} \quad (4)$$

where Δt^k is the step size and U^{k+1}, V^{k+1} are the solutions for the k-th time step. Due to the multiple temporal scales admitted by the AC/CH system, Δt^k is adaptively controlled by a method that is analogous to the switched evolution/relaxation method [5, 7]. More specifically, we start with a relatively small time step size Δt^0 and adjust its value according to

$$\begin{aligned} \Delta t^k &= \max(1/r, \min(r,s))\,\Delta t^{k-1}, \\ s &= \left(\frac{\|(\mathscr{M}_{k-1}(U^{k-1},V^{k-1}), \mathscr{N}_{k-1}(U^{k-1},V^{k-1}))^T\|_2}{\|(\mathscr{M}_k(U^k,V^k), \mathscr{N}_k(U^k,V^k))^T\|_2} \right)^p, \end{aligned} \tag{5}$$

for $k = 1,2,3,...$, where we use $r = 1.5$ and $p = 0.75$.

3 Newton-Krylov-Schwarz solver

An inexact Newton method is applied to solve the nonlinear system (4) at each time step. We denote the solution of (4) at the k-th time step as $W^{k+1} = (U^{k+1}, V^{k+1})^T$. The initial guess $X_0 = W^k$ is set to be the solution of the previous time step, then the approximate solution X_{n+1} is obtained by

$$X_{n+1} = X_n + \lambda_n S_n, \quad n = 0,1,... \tag{6}$$

Here λ_n is the steplength determined by a linesearch procedure and S_n is the Newton correction vector. To calculate S_n for each Newton iteration, a right-preconditioned linear system

$$J_n M^{-1}(MS_n) = -F_k(X_n) \tag{7}$$

is constructed and solved approximately by using a GMRES method that restarts every 30 iterations. Here $F_k(X_n) = (\mathscr{M}_k(X_n), \mathscr{N}_k(X_n))^T$ is the nonlinear residual and

$$J_n = \frac{\partial F_k(X_n)}{\partial X_n} \tag{8}$$

is the Jacobian matrix.

In (4) M^{-1} is an additive Schwarz preconditioner. We first partition Ω into np non-overlapping subdomains Ω_p, $p = 1,2,...,np$. An overlapping decomposition is obtained by extending each subdomain with δ mesh layers. Denote the overlapping subdomain as Ω_p^δ. The one-level restricted additive Schwarz (RAS, [4]) preconditioner is

$$M^{-1} = \sum_{p=1}^{np} (R_p^0)^T inv(B_p) R_p^\delta. \tag{9}$$

Here R_p^δ and $(R_p^0)^T$ serve as a restriction operator and an interpolation operator respectively; their detailed definitions can be found in [4].

In (9), $inv(B_p)$ is either an exact or approximate inverse of the subdomain problem defined by B_p. Choosing proper boundary conditions for the subdomain problems has a great impact on the convergence of the RAS preconditioner. Since the AC/CH system (1) contains two differential equations with different orders, it is natural to impose different boundary conditions. For the first equation in (1) we follow [12] by employing the following homogeneous boundary conditions

$$u = (\nabla u) \cdot \mathbf{n} = 0, \quad \partial\Omega_p^{\delta+1} \backslash \partial\Omega, \tag{10}$$

where $\mathbf{n}$ is the outward normal of $\partial\Omega_p^{\delta+1}$. For the second equation in (1), the boundary conditions are simply

$$v = 0, \quad \partial\Omega_p^{\delta} \backslash \partial\Omega. \tag{11}$$

We remark that the above boundary conditions for the subdomain problems are essential for the success of the NKS solver. Other boundary conditions are also tested but only lead to poor convergence of GMRES. To solve the subdomain problems, we use either a sparse LU factorization or a sparse incomplete LU (ILU) factorization. In doing the factorization, we use a point-block ordering for the subdomain matrix and keep the coupling between the two components at each mesh cell. Within each time step, the factorization is only done once at the first Newton iteration and is reused thereafter.

4 Numerical experiments

We carry out numerical experiments on a Dell supercomputer located at the University of Colorado Boulder. The computer consists of 1368 compute nodes, with two hex-core 2.8Ghz Intel Westmere processors and 24GB local memory in each node. Our algorithm is implemented based on the Portable, Extensible Toolkits for Scientific computations (PETSc, [1]) library. In the numerical experiments we use all 12 cores in each node and assign one subdomain per processor core. The relative stopping conditions for the Newton and GMRES iteration are respectively 1×10^{-6} and 1×10^{-5}.

4.1 Steady-state solution

The test case we study here is taken from [11]. The initial condition for the test is a randomly distributed state $(U^0, V^0) = (0.05 + \delta_u, \delta_v)$, where $\max(\|\delta_u\|_\infty, \|\delta_v\|_\infty) \leq 0.05$. The parameters are set as: $\alpha = 4$, $\beta = 2$, $\gamma = 0.005$, $\theta = 0.1$, $\rho = 0.001$.

We run the test case on a 256×256 mesh with an initial time step size $\Delta t^0 = 0.001$. The time step size is then adaptively controlled by using (5). Thanks to the fully implicit method and the adaptive time stepping strategy, we are able to obtain the steady-state solution at about $t = 100$, as seen in Figure 1 and 2. From the figures we observe that when $t < 1.4$ both the spinodal decomposition and the order-disorder type instability occur but after that the order parameter quickly tends to zero as the conserved concentration field coarsens to a stabilized state. Provided in Figure 3 is the evolution history of the time step size and the total free energy. It can be seen that by using the adaptive strategy, the time step is successfully adjusted by several orders of magnitude. The total free energy decays and finally approaches to its minimizer when the solution arrives at the steady-state.

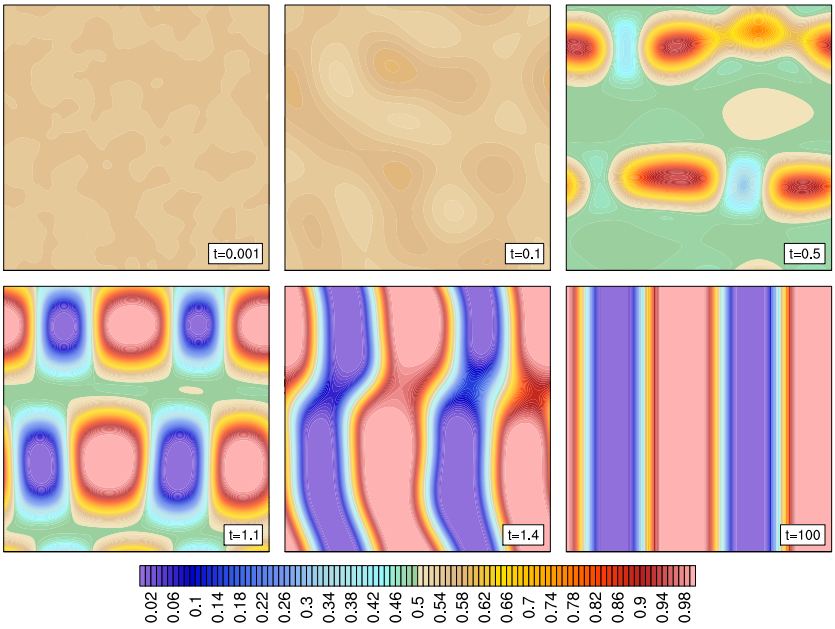


Fig. 1 Contour plots of the conserved concentration field u.

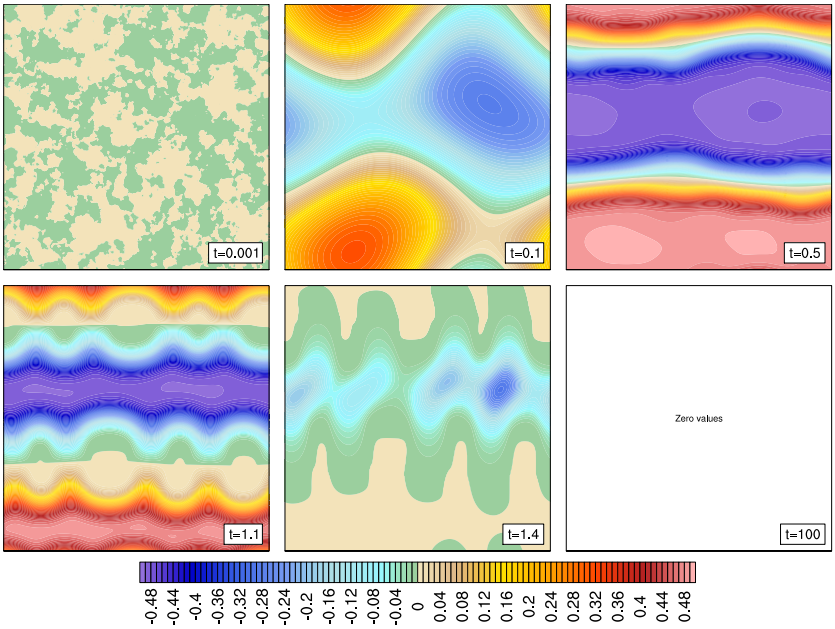


Fig. 2 Contour plots of the non-conserved order parameter v.

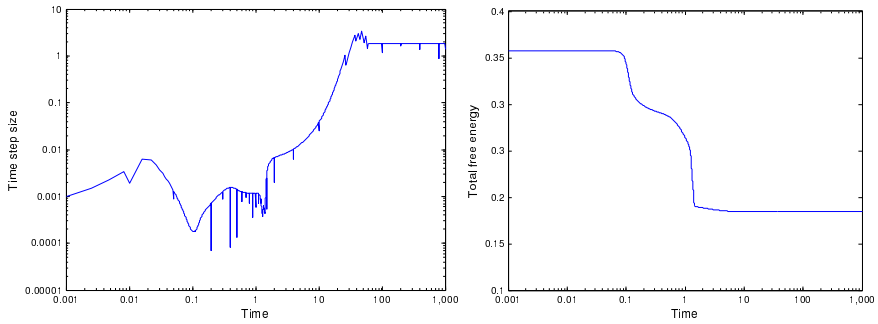


Fig. 3 Evolution history of the time step size (left panel) and the total free energy (right panel).

We remark that because of the severe stability restriction on the time step size, it is often difficult to obtain the steady-state solution when an explicit method is used. In [11], although similar tests are conducted, no steady-state solutions are obtained due to the explicit time stepping.

4.2 Parameters in the NKS solver

To understand how the parameters in the Schwarz preconditioner affect the performance, in the following experiments we run the test case on a 1152×1152 mesh with 144 processor cores by using a fixed the time step size $\Delta t = 1.0 \times 10^{-5}$ for only the first 20 steps.

We first examine the effects of different subdomain solvers. The overlap size is fixed at $\delta = 2$. In Table 1 we show the total numbers of Newton and GMRES iterations as well as the total compute time. Results for both LU and ILU with different fill-in levels are provided. From the table we find surprisingly that GMRES

Table 1 Effects of different subdomain solvers. Here "n/c" means no convergence.

	ILU(2)	ILU(4)	ILU(8)	LU	LU-blk	LU-blk-reuse
#Newton	n/c	n/c	n/c	41	41	41
#GMRES	n/c	n/c	n/c	1225	1225	1243
Time (s)	n/c	n/c	n/c	138.6	89.2	65.6

doesn't converge when ILU is the subdomain solver, even with large fill-in levels. When a sparse LU factorization is used as subdomain solver, although the point-block version doesn't change the number of iterations, the compute time is saved by around 35% compared to the non-block version. To reduce the compute time, we perform the subdomain LU factorization only once per time step, and reuse it for all the Newton iterations within the same time step. By reusing the LU factorization the

total compute time is cut by around 26% despite of the slight increase of the number of GMRES iterations. Based on the above observations, for all the following tests, we use the point-block version of sparse LU factorization and reuse the factorization within each time step.

We next investigate the performance of the NKS solver with different overlap δ. Table 2 shows the total numbers of Newton and GMRES iterations as well as the total compute time for $\delta = 0, 1, ..., 6$. It is observed from the table that: (1) the

Table 2 Results on using different overlaps.

δ	0	1	2	3	4	5	6
#Newton	41	41	41	41	41	41	41
#GMRES	21274	2482	1243	840	642	513	440
Time (s)	205.6	95.3	65.6	55.7	45.7	50.8	52.0

number of Newton iterations does not change as δ varies; (2) the number of GMRES iterations reduces when δ becomes larger; and (3) the total compute time is optimal for $\delta = 4$ in the test. Therefore we use $\delta = 4$ in our scalability tests.

4.3 Parallel scalability

In the parallel scalability tests, we fix the overlap size to be $\delta = 4$ and choose the point-block version of the sparse LU factorization (reused with each time step) as the subdomain solver. We run the tests on a 1152×1152 mesh for 20 time steps with $\Delta t = 1.0 \times 10^{-5}$ and gradually double the number of processor cores. As shown in Figure 4, when the number of processor cores is increased the total number of Newton iterations stays unchanged while the total number of GMRES iterations increases slightly. Further from Figure 5 we observe that the total compute time is reduced almost linearly as more processor cores are used. A total of 12.35 speedup is achieved when the number of processor cores increases from 144 to 2304, leading to a parallel efficiency of 78.1%.

References

1. Balay, S., Brown, J., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.3, Argonne National Laboratory (2012)
2. Barrett, J.W., Blowey, J.F.: Finite element approximation of a degenerate Allen-Cahn/Cahn-Hilliard system. SIAM J. Numer. Anal. **39**, 1598–1624 (2002)
3. Cahn, J., Novick-Cohen, A.: Evolution equations for phase separation and ordering in binary alloys. J. Statist. Phys. **76**, 877–909 (1994)

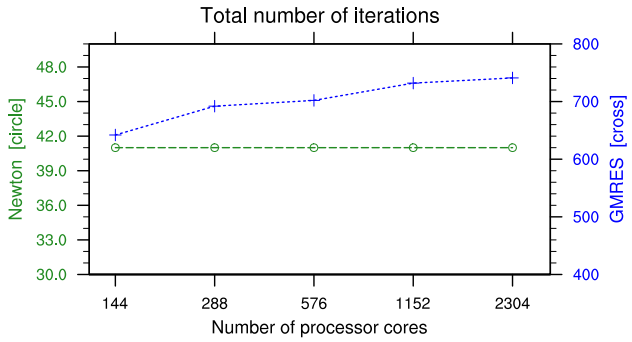


Fig. 4 Total numbers of Newton and GMRES iterations for the first 20 time steps.

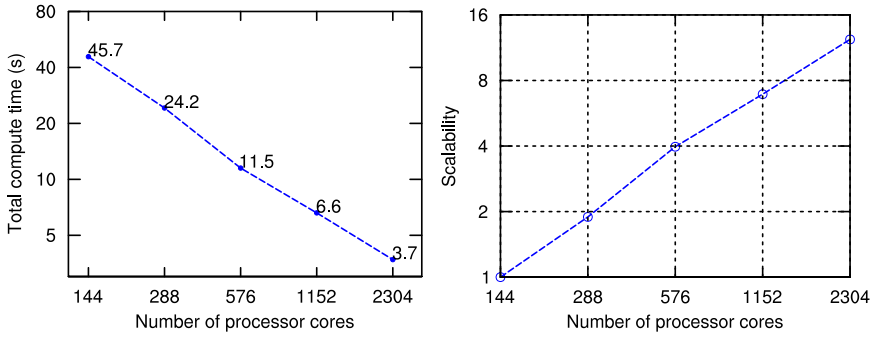


Fig. 5 Total compute time (left) and parallel scalability (right) results.

4. Cai, X.-C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J. Sci. Comput. **21**, 792–797 (1999)
5. Gropp, W.D., Kaushik, D.K., Keyes, D.E., Smith, B.: Performance modeling and tuning of an unstructured mesh CFD application. In: Proc. Supercomputing 2000. IEEE Computer Society (2000)
6. Millett, P.C., Rokkam, S., El-Azab, A., Tonks, M., Wolf, D.: Void nucleation and growth in irradiated polycrystalline metals: a phase-field model. Modelling Simul. Mater. Sci. Eng. **17**, 064,003 (2009)
7. Mulder, W.A., Leer, B.V.: Experiments with implicit upwind methods for the Euler equations. J. Comput. Phys. **59**, 232–246 (1985)
8. Rokkam, S., El-Azab, A., Millett, P., Wolf, D.: Phase field modeling of void nucleation and growth in irradiated metals. Modelling Simul. Mater. Sci. Eng. **17**, 064,002 (2009)
9. Tonks, M.R., Gaston, D., Millett, P.C., Andrs, D., Talbot, P.: An object-oriented finite element framework for multiphysics phase field simulations. Comput. Mater. Sci. **51**, 20 – 29 (2012)
10. Wang, L., Lee, J., Anitescu, M., Azab, A.E., Mcinnes, L.C., Munson, T., Smith, B.: A differential variational inequality approach for the simulation of heterogeneous materials. In: Proc. SciDAC 2011 (2011)
11. Xia, Y., Xu, Y., Shu, C.W.: Application of the local discontinuous Galerkin method for the Allen-Cahn/Cahn-Hilliard system. Commun. Comput. Phys. **5**, 821–835 (2009)
12. Yang, C., Cai, X.-C., Keyes, D.E., Pernice, M.: Parallel domain decomposition methods for the 3D Cahn-Hilliard equation. In: Proc. SciDAC 2011 (2011)

Surrogate Functional Based Subspace Correction Methods for Image Processing

Michael Hintermüller[1] and Andreas Langer[2]

1 Introduction

Recently in [4, 5, 6] subspace correction methods for non-smooth and non-additive problems have been introduced in the context of image processing, where the non-smooth and non-additive total variation (TV) plays a fundamental role as a regularization technique, since it preserves edges and discontinuities in images. We recall, that for $u \in L^1(\Omega)$, $V(u,\Omega) := \sup\left\{\int_\Omega u \mathrm{div}\phi dx : \phi \in [C_c^1(\Omega)]^2, \|\phi\|_\infty \leq 1\right\}$ is the variation of u. In the event that $V(u,\Omega) < \infty$ we denote $|Du|(\Omega) = V(u,\Omega)$ and call it the total variation of u in Ω [1].

In this paper, as in [6], we consider functionals, which consist of a non-smooth and non-additive regularization term and a weighted combination of an ℓ^1-term and a quadratic ℓ^2-term; see (1) below. This type of functional has been shown to be particularly efficient to eliminate simultaneously Gaussian and salt-and-pepper noise. In [6] an estimate of the distance of the limit point obtained from the proposed subspace correction method to the global minimizer is established. In that paper the exact subspace minimization problems are minimized, which are in general not easily solved. Therefore, in the present paper we analyse a subspace correction approach in which the subproblems are approximated by so-called *surrogate* functionals, as in [4, 5]. In this situation, as in [6], we are able to achieve an estimate for the distance of the computed solution to the real global minimizer. With the help of this estimate we show in our numerical experiments that the proposed algorithm generates a sequence which converges to the expected minimizer.

2 Notations

For the sake of brevity we consider a two dimensional setting only. We define $\Omega = \{x_1 < \ldots < x_N\} \times \{y_1 < \ldots < y_N\} \subset \mathbb{R}^2$, and $H = \mathbb{R}^{N\times N}$, where $N \in \mathbb{N}$. For $u \in H$ we write $u = u(x) = u(x_i, y_j)$, where $i, j \in \{1,\ldots,N\}$ and $x \in \Omega$. Let $h = x_{i+1} - x_i = y_{j+1} - y_j$ be the equidistant step-size. We define the scalar product of $u, v \in H$ by $\langle u, v\rangle_H = h^2 \sum_{x\in\Omega} u(x)v(x)$ and the scalar product of $p, q \in H^2$ by $\langle p, q\rangle_{H^2} = h^2 \sum_{x\in\Omega} \langle p(x), q(x)\rangle_{\mathbb{R}^2}$ with $\langle z, w\rangle_{\mathbb{R}^2} = \sum_{j=1}^2 z_j w_j$ for every $z = (z_1, z_2) \in \mathbb{R}^2$ and $w =$

[1] Department of Mathematics, Humboldt-University of Berlin, Unter den Linden 6, 10099 Berlin, Germany, e-mail: hint@math.hu-berlin.de · [2] Institute for Mathematics and Scientific Computing, University of Graz, Heinrichstraße 36, A-8010 Graz, Austria, e-mail: andreas.langer@uni-graz.at

$(w_1,w_2)\in\mathbb{R}^2$. We also use $\|u\|_{\ell^p(\Omega)}=\left(h^2\sum_{x\in\Omega}|u(x)|^p\right)^{1/p}$, $1\le p<\infty$, $\|u\|_{\ell^\infty(\Omega)}=\sup_{x\in\Omega}|u(x)|$ and $\|\cdot\|$, when any norm can be taken.

The discrete gradient ∇u is denoted by $(\nabla u)(x)=((\nabla u)^1(x),(\nabla u)^2(x))$ with $(\nabla u)^1(x)=\frac{1}{h}(u(x_{i+1},y_j)-u(x_i,y_j))$ if $i<N$ and $(\nabla u)^1(x)=0$ if $i=N$, and $(\nabla u)^2(x)=\frac{1}{h}(u(x_i,y_{j+1})-u(x_i,y_j))$ if $j<N$ and $(\nabla u)^2(x)=0$ if $j=N$, for all $x\in\Omega$. For $\omega\in H^2$ we define $\varphi:\mathbb{R}\to\mathbb{R}$ by $\varphi(|\omega|)(\Omega):=h^2\sum_{x\in\Omega}\varphi(|\omega(x)|)$, where $|z|=\sqrt{z_1^2+z_2^2}$. In particular we define the *total variation* of u by setting $\varphi(t)=t$ and $\omega=\nabla u$, i.e., $|\nabla u|(\Omega):=h^2\sum_{x\in\Omega}|\nabla u(x)|$.

For an operator T we denote by T^* its adjoint. Further we introduce the *discrete divergence* $\mathrm{div}:H^2\to H$ defined by $\mathrm{div}=-\nabla^*$ (∇^* is the adjoint of the gradient ∇), in analogy to the continuous setting. The symbol 1 indicates the constant vector with entry values 1 and 1_D is the characteristic function of $D\subset\Omega$.

For a convex functional $J:H\to\bar{\mathbb{R}}$, we define the *subdifferential* of J at $v\in H$ as the set valued mapping $\partial J(v):=\emptyset$ if $J(v)=\infty$ and $\partial J(v):=\{v^*\in H:\langle v^*,u-v\rangle_H+J(v)\le J(u)\quad\forall u\in H\}$ otherwise. It is clear from this definition that $0\in\partial J(v)$ if and only if v is a minimizer of J. Whenever the underlying space is important, then we write $\partial_{V_i}J$ or $\partial_H J$.

3 Subspace Correction Approaches

As in [6] we are interested in minimizing by means of subspace correction the following functional

$$J(u)=\alpha_S\|Su-g_S\|_{\ell^1(\Omega)}+\alpha_T\|Tu-g_T\|^2_{\ell^2(\Omega)}+\varphi(|\nabla u|)(\Omega),\tag{1}$$

where $S,T:H\to H$ are bounded linear operators, $g_S,g_T\in H$ are given data, and $\alpha_S,\alpha_T\ge 0$ with $\alpha_S+\alpha_T\ge\tau>0$. We assume that J is bounded from below and coercive, i.e., $\{u\in H\,:\,J(u)\le C\}$ is bounded in H for all constants $C>0$, in order to guarantee that (1) has minimizers. Moreover we assume that $\varphi:\mathbb{R}\to\mathbb{R}$ is a convex function, nondecreasing in $\mathbb{R}^+$ with (i) $\varphi(0)=0$ and (ii) $cz-b\le\varphi(z)\le cz+b$, for all $z\in\mathbb{R}^+$ for some constant $c>0$ and $b\ge 0$.

Note that for the particular example $\varphi(t)=t$, the third term in (1) becomes the well-known total variation of u in Ω and we call (1) the L^1-L^2-TV model.

Now we seek to minimize (1) by decomposing H into two subspaces V_1 and V_2 such that $H=V_1+V_2$. Note that a generalization to multiple splittings can be performed straightforwardly. However, here we will restrict ourselves to a decomposition into two domains only for simplicity. By V_i^c we denote the orthogonal complement of V_i in H and we define by $\pi_{V_i^c}$ the corresponding orthogonal projection onto V_i^c for $i=1,2$.

With this splitting we want to minimize J by suitable instances of the following alternating algorithm:

Choose an initial $u^{(0)}=:u_1^{(0)}+u_2^{(0)}\in V_1+V_2$, for example, $u^{(0)}=0$, and iterate

$$\begin{cases} u_1^{(n+1)} = \arg\min\limits_{u_1 \in V_1} J(u_1 + u_2^{(n)}), \\ u_2^{(n+1)} = \arg\min\limits_{u_2 \in V_2} J(u_1^{(n+1)} + u_2), \\ u^{(n+1)} := u_1^{(n+1)} + u_2^{(n+1)}. \end{cases} \tag{2}$$

Differently from the case in [6], where the authors solved the exact subspace minimization problems in (2), we suggest now to approximate the subdomain problems by so-called surrogate functionals (cf. [2, 3, 4, 5, 8]): Assume $a, u_i \in V_i$, $u_{-i} \in V_{-i}$, and define

$$\begin{aligned} J^s(u_i + u_{-i}, a + u_{-i}) := J(u_i + u_{-i}) + \alpha_T \big(& \delta \|u_i + u_{-i} - (a + u_{-i})\|^2_{\ell^2(\Omega)} \\ & - \|T(u_i + u_{-i} - (a + u_{-i}))\|^2_{\ell^2(\Omega)} \big) \\ = \; J(u_i + u_{-i}) + \alpha_T & \left(\delta \|u_i - a\|^2_{\ell^2(\Omega)} - \|T(u_i - a)\|^2_{\ell^2(\Omega)} \right) \end{aligned} \tag{3}$$

for $i = 1, 2$ and $-i \in \{1, 2\} \setminus \{i\}$, where $\delta > \|T\|^2$. Then an approximate solution to $\min_{u_i \in V_i} J(u_1 + u_2)$ is realized by using the following algorithm: For $u_i^{(0)} \in V_i$,

$$u_i^{(\ell+1)} = \arg\min_{u_i \in V_i} J^s(u_i + u_{-i}, u_i^{(\ell)} + u_{-i}), \quad \ell \geq 0,$$

where $u_{-i} \in V_{-i}$ for $i = 1, 2$ and $-i \in \{1, 2\} \setminus \{i\}$.

The alternating domain decomposition algorithm reads then as follows:

Choose an initial $u^{(0)} =: \tilde{u}_1^{(0)} + \tilde{u}_2^{(0)} \in V_1 + V_2$, for example, $u^{(0)} = 0$, and iterate

$$\begin{cases} \begin{cases} u_1^{(n+1,0)} = \tilde{u}_1^{(n)}, \\ u_1^{(n+1,\ell+1)} = \arg\min\limits_{u_1 \in V_1} J^s(u_1 + \tilde{u}_2^{(n)}, u_1^{(n+1,\ell)} + \tilde{u}_2^{(n)}), \; \ell = 0, \ldots, L-1, \end{cases} \\ \begin{cases} u_2^{(n+1,0)} = \tilde{u}_2^{(n)}, \\ u_2^{(n+1,m+1)} = \arg\min\limits_{u_2 \in V_2} J^s(u_1^{(n+1,L)} + u_2, u_2^{(n+1,m)} + u_1^{(n+1,L)}), \; m = 0, \ldots, M-1, \end{cases} \\ u^{(n+1)} := u_1^{(n+1,L)} + u_2^{(n+1,M)}, \; \tilde{u}_1^{(n+1)} = \chi_1 \cdot u^{(n+1)}, \; \tilde{u}_2^{(n+1)} = \chi_2 \cdot u^{(n+1)}, \end{cases} \tag{4}$$

where $\chi_1, \chi_2 \in H$ have the properties (i) $\chi_1 + \chi_2 = 1$ and (ii) $\chi_i \in V_i$ for $i = 1, 2$. Let $\kappa := \max\{\|\chi_1\|_\infty, \|\chi_2\|_\infty\} < \infty$.

The parallel version of the algorithm in (4) reads as follows:

Choose an initial $u^{(0)} =: \tilde{u}_1^{(0)} + \tilde{u}_2^{(0)} \in V_1 + V_2$, for example, $u^{(0)} = 0$, and iterate

$$\begin{cases} \begin{cases} u_1^{(n+1,0)} = \tilde{u}_1^{(n)}, \\ u_1^{(n+1,\ell+1)} = \arg\min\limits_{u_1 \in V_1} J^s(u_1 + \tilde{u}_2^{(n)}, u_1^{(n+1,\ell)} + \tilde{u}_2^{(n)}),\ \ell = 0,\dots,L-1, \end{cases} \\ \begin{cases} u_2^{(n+1,0)} = \tilde{u}_2^{(n)}, \\ u_2^{(n+1,m+1)} = \arg\min\limits_{u_2 \in V_2} J^s(\tilde{u}_1^{(n)} + u_2, u_2^{(n+1,m)} + \tilde{u}_1^{(n)}),\ m = 0,\dots,M-1, \end{cases} \\ u^{(n+1)} := \frac{u_1^{(n+1,L)} + u_2^{(n+1,M)} + u^{(n)}}{2},\ \tilde{u}_1^{(n+1)} = \chi_1 \cdot u^{(n+1)},\ \tilde{u}_2^{(n+1)} = \chi_2 \cdot u^{(n+1)}. \end{cases} \tag{5}$$

Note that we prescribe a finite number L and M of inner iterations for each subspace, respectively. Hence we do not get a minimizer of the original subspace minimization problems in (2), but approximations of such minimizers. Moreover, observe that $u^{(n+1)} = \tilde{u}_1^{(n+1)} + \tilde{u}_2^{(n+1)}$, with $u_i^{(n+1,L)} \neq \tilde{u}_i^{(n+1)}$, for $i = 1,2$, in general.

We have that $u_1^{(n+1,L)} \in \arg\min_{u \in H} \left\{ J^s(u + \tilde{u}_2^{(n)}, u_1^{(n+1,L-1)} + \tilde{u}_2^{(n)}) : \pi_{V_1^c} u = 0 \right\}$. Then, by [7, Theorem 2.1.4, p. 305] there exists an $\eta_1^{(n+1)} \in \mathrm{Range}(\pi_{V_1^c})^* \simeq V_1^c$ such that

$$0 \in \partial_H J^s(\cdot + \tilde{u}_2^{(n)}, u_1^{(n+1,L-1)} + \tilde{u}_2^{(n)})(u_1^{(n+1,L)}) + \eta_1^{(n+1)}. \tag{6}$$

Analogously, we have that if $u_2^{(n+1,M)}$ is a minimizer of the second optimization problem in (4) or (5), then there exists an $\eta_2^{(n+1)} \in \mathrm{Range}(\pi_{V_2^c})^* \simeq V_2^c$ such that

$$0 \in \partial_H J^s(u_1^{(n+1,L)} + \cdot, u_1^{(n+1,L)} + \tilde{u}_2^{(n+1,M-1)})(u_2^{(n+1,M)}) + \eta_2^{(n+1)}, \text{ or} \tag{7}$$

$$0 \in \partial_H J^s(\tilde{u}_1^{(n,L)} + \cdot, \tilde{u}_1^{(n,L)} + \tilde{u}_2^{(n+1,M-1)})(u_2^{(n+1,M)}) + \eta_2^{(n+1)}, \tag{8}$$

respectively.

3.1 Convergence Properties

In this section we state convergence properties of the subspace correction methods in (4) and (5). In particular, the following three propositions are direct consequences of statements in [4, 5, 6].

Proposition 1. *The algorithms in (4) and (5) produce a sequence $(u^{(n)})_n$ in H with the following properties:*

(i) $J(u^{(n)}) > J(u^{(n+1)})$ for all $n \in \mathbb{N}$ (unless $u^{(n)} = u^{(n+1)}$);
(ii) $\lim_{n\to\infty} \|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\|_{\ell^2(\Omega)} = 0$ and $\lim_{n\to\infty} \|u_2^{(n+1,m+1)} - u_2^{(n+1,m)}\|_{\ell^2(\Omega)} = 0$ for all $\ell \in \{0,\dots,L-1\}$ and $m \in \{0,\dots,M-1\}$;
(iii) $\lim_{n\to\infty} \|u^{(n+1)} - u^{(n)}\|_{\ell^2(\Omega)} = 0$;
(iv) the sequence $(u^{(n)})_n$ has subsequences that converge in H.

The proof of this proposition is analogous to the one in [5, Theorem 5.1].

Proposition 2. *The sequences $(\tilde{u}_i^{(n)})_n$ for $i = 1,2$ generated by the algorithm in (4) or (5) are bounded in H and hence have accumulation points $\tilde{u}_i^{(\infty)}$, respectively.*

Proof. By the boundedness of the sequence $(u^{(n)})_n$ we obtain $\|\tilde{u}_i^{(n)}\| = \|\chi_i u^{(n)}\| \leq \kappa\|u^{(n)}\| \leq C < \infty$ and hence $(\tilde{u}_i^{(n)})_n$ is bounded for $i = 1,2$. □

Remark 1. From the previous proposition it directly follows by the coercivity assumption on J that the sequences $(u_1^{(n,\ell)})_n$ and $(u_2^{(n,m)})_n$ are bounded for all $\ell \in \{0,\dots,L\}$ and $m \in \{0,\dots,M\}$.

Proposition 3. *Let $u_1^{(\infty)}$, $u_2^{(\infty)}$, and $\tilde{u}_i^{(\infty)}$ be accumulation points of the sequences $(u_1^{(n,L)})_n$, $(u_2^{(n,M)})_n$, and $(\tilde{u}_i^{(n)})_n$ generated by the algorithms in (4) and (5), then $u_i^{(\infty)} = \tilde{u}_i^{(\infty)}$, for $i = 1,2$.*

One shows this statement analogous to the first part of the proof of [4, Theorem 5.7].

Moreover, as in [6] we are able to establish an estimate of the distance of the limit point obtained from the subspace correction method to the true global minimizer.

Theorem 1. *Let $\alpha_S \geq \tau$, u^* a minimizer of J, and $u^{(\infty)}$ an accumulation point of the sequence $(u^{(n)})_n$ generated by the algorithm in (4) or (5). Then we have that*

(i) $u^{(\infty)}$ is a minimizer of J or

(ii) there exists a constant $\beta > 0$ (independent of α_T) such that $\|u^{(\infty)} - u^*\|_{\ell^2(\Omega)} \leq \beta$ or

(iii) if $\alpha_T < \frac{\gamma}{\beta^2\delta}$ for $0 < \gamma \leq J(u^{(\infty)}) - J(u^*)$, then $\|u^{(\infty)} - u^*\|_{\ell^2(\Omega)} \leq \frac{\beta^2\|\hat{\eta}\|_{\ell^2(\Omega)}}{\gamma - \alpha_T\delta\beta^2}$, where $\|\hat{\eta}\|_{\ell^2(\Omega)} = \min\{\|\eta_1^{(\infty)}\|_{\ell^2(\Omega)}, \|\eta_2^{(\infty)}\|_{\ell^2(\Omega)}\}$ and $\eta_i^{(\infty)}$ is an accumulation point of the sequence $(\eta_i^{(n)})_n$ for $i = 1,2$ defined as in (6)-(8) respectively, or

(iv) if T^*T is positive definite with smallest Eigenvalue $\sigma > 0$, $\alpha_T > 0$ and $\|T\|^2 < \delta < 2\sigma$, then we have $\|u^* - u^{(\infty)}\|_{\ell^2(\Omega)} \leq \frac{\|\hat{\eta}\|_{\ell^2(\Omega)}}{\alpha_T(2\sigma - \delta)}$.

Proof. Since $(u_1^{(n+1,L)})_n$, $(u_1^{(n+1,L-1)})_n$, and $(\tilde{u}_2^{(n)})_n$ are bounded and based on the fact that $\partial J^s(\xi,\tilde{\xi})$ is compact for any $\xi,\tilde{\xi} \in H$ we obtain that $(\eta_1^{(n)})_n$ is bounded, cf. [6, Corollary 4.7]. By noting that $(u_1^{(n+1,L)})_n$ and $(u_1^{(n+1,L-1)})_n$ have the same limit for $n \to \infty$, see Proposition 1, we subtract a suitable subsequence $(n_k)_k$ with limits $\eta_1^{(\infty)}$, $u_1^{(\infty)}$, and $\tilde{u}_2^{(\infty)}$ such that (6)-(8) respectively are still valid, cf. [9, Theorem 24.4, p 233], i.e., $0 \in \partial_H J^s(\cdot + \tilde{u}_2^{(\infty)}, u_1^{(\infty)} + \tilde{u}_2^{(\infty)})(u_1^{(\infty)}) + \eta_1^{(\infty)}$. By the definition of the subdifferential and Proposition 3 we obtain $J(u^{(\infty)}) = J^s(u^{(\infty)}, u^{(\infty)}) \leq J^s(v, u^{(\infty)}) + \langle \eta_1^{(\infty)}, u^{(\infty)} - v\rangle_H \leq J^s(v, u^{(\infty)}) + \|\eta_1^{(\infty)}\|_{\ell^2(\Omega)}\|u^{(\infty)} - v\|_{\ell^2(\Omega)}$ for all $v \in H$. Similarly one can show that $J(u^{(\infty)}) \leq J^s(v, u^{(\infty)}) + \|\eta_2^{(\infty)}\|_{\ell^2(\Omega)}\|u^{(\infty)} - v\|_{\ell^2(\Omega)}$ for all $v \in H$, and hence we have

$$J(u^{(\infty)}) \leq J^s(v, u^{(\infty)}) + \|\hat{\eta}\|_{\ell^2(\Omega)}\|u^{(\infty)} - v\|_{\ell^2(\Omega)} \tag{9}$$

for all $v \in H$, where $\|\hat{\eta}\|_{\ell^2(\Omega)} = \min\{\|\eta_1^{(\infty)}\|_{\ell^2(\Omega)}, \|\eta_2^{(\infty)}\|_{\ell^2(\Omega)}\}$.

Let $u^* \in \arg\min_{u \in H} J(u)$. Then there exists a $\rho \geq 0$ such that $J(u^{(\infty)}) = J(u^*) + \rho$.

(i) If $\rho = 0$, then it immediately follows that $u^{(\infty)}$ is a minimizer of J.

(ii) If $\rho > 0$, then from the coercivity condition we obtain that there exists a constant $\beta > 0$, independent of α_T, such that $\|u^{(\infty)} - u^*\|_{\ell^2(\Omega)} \leq \beta < +\infty$.

(iii) If $\alpha_T < \frac{\gamma}{\beta^2 \delta}$ for $0 < \gamma \leq J(u^{(\infty)}) - J(u^*)$, then $J(u^{(\infty)}) \geq J(u^*) + \frac{\gamma}{\beta^2}\|u^{(\infty)} - u^*\|^2_{\ell^2(\Omega)}$. Setting $v = u^*$ in (9) and using the last inequality we obtain

$$\alpha_T \left(\delta \|u^* - u^{(\infty)}\|^2_{\ell^2(\Omega)} - \|T(u^* - u^{(\infty)})\|^2_{\ell^2(\Omega)} \right) + \|\hat{\eta}\|_{\ell^2(\Omega)} \|u^{(\infty)} - u^*\|_{\ell^2(\Omega)} \geq \frac{\gamma}{\beta^2} \|u^{(\infty)} - u^*\|^2_{\ell^2(\Omega)}. \quad (10)$$

From the latter inequality we get $\|\hat{\eta}\|_2 \geq (\frac{\gamma}{\beta^2} - \alpha_T \delta)\|u^{(\infty)} - u^*\|_{\ell^2(\Omega)}$ and since $\alpha_T \delta < \frac{\gamma}{\beta^2}$ we obtain $\frac{\beta^2 \|\hat{\eta}\|_{\ell^2(\Omega)}}{\gamma - \alpha_T \delta \beta^2} \geq \|u^{(\infty)} - u^*\|_{\ell^2(\Omega)}$.

(iv) If $\alpha_T > 0$ and T^*T is symmetric positive definite with smallest Eigenvalue $\sigma > 0$, then the factor $\frac{\gamma}{\beta^2}$ on the right hand side of the inequality in (10) is replaced by $\alpha_T \sigma$, cf. [6], and (10) reads as follows

$$\alpha_T(\sigma - \delta)\|u^* - u^{(\infty)}\|^2_{\ell^2(\Omega)} + \alpha_T \|T(u^* - u^{(\infty)})\|^2_{\ell^2(\Omega)} \leq \|\hat{\eta}\|_{\ell^2(\Omega)} \|u^{(\infty)} - u^*\|_{\ell^2(\Omega)}.$$

By using once more the symmetric positive definiteness assumption on T^*T we obtain from the latter inequality that $\alpha_T(2\sigma - \delta)\|u^* - u^{(\infty)}\|^2_{\ell^2(\Omega)} \leq \|\hat{\eta}\|_{\ell^2(\Omega)} \|u^{(\infty)} - u^*\|_{\ell^2(\Omega)}$. If $2\sigma > \delta$ then we get $\|u^* - u^{(\infty)}\|_{\ell^2(\Omega)} \leq \frac{\|\hat{\eta}\|_{\ell^2(\Omega)}}{\alpha_T(2\sigma - \delta)}$.

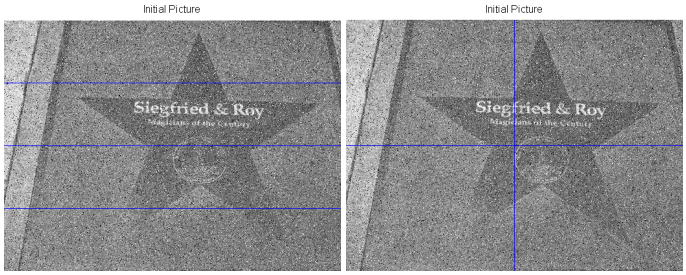
□

4 Numerical Experiments

We present numerical experiments obtained by the parallel algorithm in (5) for the application of image deblurring, i.e., $S = T$ are blurring operators and $\varphi(|\nabla u|)(\Omega) = |\nabla u|(\Omega)$ (the total variation of u in Ω). The minimization problems of the subdomains are implemented in the same way as described in [6] by noting that the functional to be considered in each subdomain is now the strictly convex functional in (3).

We consider an image of size 1920×2576 pixels which is corrupted by Gaussian blur with kernel size 15×15 pixels and standard deviation 2. Additionally 4% salt-and-pepper noise (i.e., 4% of the pixels are either flipped to black or white) and Gaussian white noise with zero mean and variance 0.01 is added.

In order to show the efficiency of the parallel algorithm in (5) for decomposing the spatial domain into subomains, we compare its performance with the L^1-L^2-TV algorithm presented in [6], which solves the problem on all of Ω without any splitting. We consider splittings of the domain in stripes, cf. Figure 1(a), and in windows as depicted in Figure 1(b) for different numbers of subdomains ($D = 4, 16, 64$).



(a) (b)

Fig. 1 Image of size 1920×2576 pixels which is corrupted by Gaussian blur with kernel size 15×15 pixels and standard deviation 2, 4% salt-and-pepper noise, and Gaussian white noise with zero mean and variance 0.01. In (a) decomposition of the spatial domain into stripes and in (b) into windows.

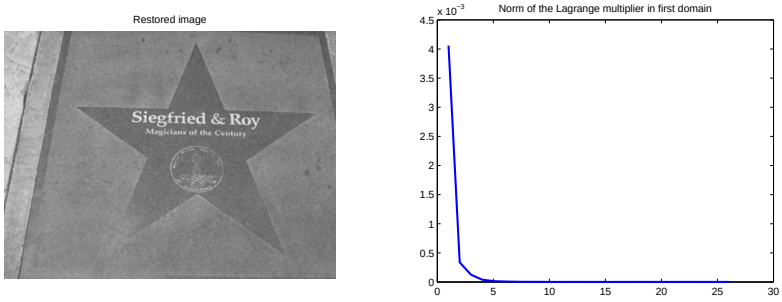
The algorithms are stopped as soon as the energy J reaches a significance level J^*, i.e., when $J(u^{(n)}) \leq J^*$ for the first time. For reason of comparison we experimentally choose $J^* = 0.059054$, i.e., we once restored the image of interest until we observed a visually satisfying restoration and the associated energy-value as J^*. In the subspace correction algorithm as well as in the L^1-L^2-TV algorithm we restore the image by setting $\alpha_S = 0.5$, $\alpha_T = 0.4$, and $\delta = 1.1$. The computations are done in Matlab on a computer with 256 cores and the multithreading-option is activated.

Table 1 presents the computational time and number of iterations the algorithms need to fulfill the stopping criterion for different number of subdomains. We clearly see that the domain decomposition algorithm for $D = 4, 16, 64$ subdomains is much faster than the L^1-L^2-TV algorithm ($D = 1$). Since a blurring operator is in general non-local, in each iteration $u^{(n)}$ has been communicated to each subdomain. Therefore the communication time becomes substantial for splittings into 16 or more domains such that the algorithm needs more time to reach the stopping criterion.

Table 1 Restoration of the image in Figure 1: Computational performance (CPU time in seconds and the number of iterations) for the global L^1-L^2-TV algorithm and for the parallel domain decomposition algorithms with $\alpha_1 = 0.5$, $\alpha_2 = 0.4$ for different numbers of subdomains ($D = 4, 16, 64$).

# Domains	window-splitting	stripe-splitting
$D = 1$ (L^1-L^2-TV alg.): 11944 s / 131 it		
$D = 4$:	2374 s / 27 it	2340 s / 27 it
$D = 16$:	2914 s / 27 it	2982 s / 27 it
$D = 64$:	7833 s / 27 it	8797 s / 28 it

In Figure 2 we depict the progress of the minimal Lagrange multiplier $\eta^{(n)} := \min_i \{\|\eta_i^{(n)}\|_{\ell^2(\Omega)}\}$, which indicates that the parallel algorithm indeed converges to a minimizer of the functional J.



(a) (b)

Fig. 2 (a) Restoration of the image in Figure 2 by the parallel subspace correction algorithm in (5). (b) The progress of the minimal Lagrange multiplier $\eta^{(n)}$.

Acknowledgements This work was supported by the Austrian Science Fund FWF through the START Project Y 305-N18 “Interfaces and Free Boundaries” and the SFB Project F32 04-N18 “Mathematical Optimization and Its Applications in Biomedical Sciences” as well as by the German Research Fund (DFG) through the Research Center MATHEON Project C28 and the SPP 1253 “Optimization with Partial Differential Equations”. M.H. also acknowledges support through a J. Tinsely Oden Fellowship at the Institute for Computational Engineering and Sciences (ICES) at UT Austin, Texas, USA.

References

1. Ambrosio, L., Fusco, N., D., P.: Functions of Bounded Variation and Free Discontinuity Problems. Oxford Mathematical Monographs. Oxford: Clarendon Press. xviii, Oxford (2000)
2. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul. **4**, 1168–1200 (2005)
3. Fornasier, M., Kim, Y., Langer, A., Schönlieb, C.B.: Wavelet decomposition method for l_2/tv-image deblurring. SIAM J. Imaging Sciences **5**, 857–885 (2012)
4. Fornasier, M., Langer, A., Schönlieb, C.B.: A convergent overlapping domain decomposition method for total variation minimization. Numer. Math. **116**, 645–685 (2010)
5. Fornasier, M., Schönlieb, C.B.: Subspace correction methods for total variation and ℓ_1-minimization. SIAM J. Numer. Anal. **47**, 3397–3428 (2009)
6. Hintermüller, M., Langer, A.: Subspace correction methods for non-smooth and non-additive convex variational problems in image processing. SFB-Report 2012-021, Institute for Mathematics and Scientific Computing, Karl-Franzens-University Graz (2012)
7. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms I, Vol. 305 of Grundlehren der mathematischen Wissenschaften. Springer-Verlag, Berlin (1996)
8. Nesterov, Y.: Gradient methods for minimizing composite objective function. CORE Discussion Paper 2007/76 (2007)
9. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princton, NJ (1970)

Practical aspects of domain decomposition in Jacobi-Davidson for parallel performance

Menno Genseberger[1]

1 Introduction

Most computational work in Jacobi-Davidson [7], an iterative method for large scale eigenvalue problems, is due to a so-called correction equation. For this, to reduce wall clock time and local memory requirements, [3, 5] proposed a domain decomposition strategy that was further improved in [4] (§ 2 and § 3).

Here we investigate practical aspects for parallel performance of the strategy by scaling experiments on supercomputers (§ 4). This is of interest for large scale eigenvalue problems that need a massively parallel treatment.

2 Domain decomposition

In [3, 5] a domain decomposition preconditioning technique for the (approximate) solution of the correction equation was proposed. This technique is based on a nonoverlapping additive Schwarz method with locally optimized coupling parameters by Tan & Borsboom [8, 9] (belonging to the class of optimized Schwarz methods [2]).

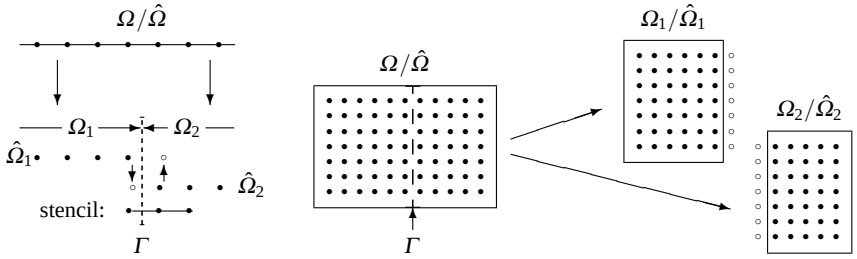


Fig. 1 Decomposition in one (left picture) and two dimensions (right picture).

For some partial differential equation (PDE) defined on a domain Ω with appropriate boundary conditions, Ω is covered by a grid $\hat{\Omega}$ and the PDE is discretized accordingly, with unknowns defined on the grid points, yielding the linear system

$$\mathbf{B}\mathbf{y} = \mathbf{d}. \tag{1}$$

[1] Deltares, PO Box 177, 2600 MH Delft, The Netherlands, e-mail: Menno.Genseberger@deltares.nl

Now, the domain decomposition technique
1. Enhances the linear system (1) into $\mathbf{B}_C\,\mathbf{y}_{\approx} = \mathbf{d}_0$ with the following structure

$$\begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{1\ell} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{\ell 1} & B_{\ell\ell} & B_{\ell r} & 0 & 0 & \mathbf{0} \\ \mathbf{0} & C_{\ell\ell} & C_{\ell r} & -C_{\ell\ell} & -C_{\ell r} & \mathbf{0} \\ \mathbf{0} & -C_{r\ell} & -C_{rr} & C_{r\ell} & C_{rr} & \mathbf{0} \\ \mathbf{0} & 0 & 0 & B_{r\ell} & B_{rr} & \mathbf{B}_{r2} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{2r} & \mathbf{B}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ y_\ell \\ \tilde{y}_r \\ \tilde{y}_\ell \\ y_r \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ d_\ell \\ 0 \\ 0 \\ d_r \\ \mathbf{d}_2 \end{bmatrix} \tag{2}$$

in case of a two subdomain decomposition (generalization is straightforward). Here Ω is decomposed in two nonoverlapping subdomains Ω_1 and Ω_2 with interface (or internal boundary) Γ (see Fig. 1). The subdomains are covered by subgrids $\hat{\Omega}_1$ and $\hat{\Omega}_2$ with additional grid points located just outside the subdomain near the interface Γ (the open bullets "$\circ$" in Fig. 1) such that no splitting of the original discretized operator (or stencil) has to be made. For $\mathbf{B}$, the labels $1, 2, \ell$, and r, respectively, refer to operations on data from/to subdomain Ω_1, Ω_2, and left, right from the interface Γ, respectively. For $\mathbf{y}$ and $\mathbf{d}$, the labels $1, 2, \ell$, and r, respectively, refer to data in subdomain Ω_1, Ω_2, and left, right from the interface Γ, respectively. Here, subvector y_ℓ (y_r respectively) contains those unknowns on the left (right) from Γ that are coupled by the stencil both with unknowns in Ω_1 (Ω_2) and unknowns on the right (left) from Γ. Subvector $\tilde{y}_r$ ($\tilde{y}_\ell$ respectively) contains the unknowns at the additional grid points of the subgrid for Ω_1 (Ω_2) on the right (left) of Γ. For the unknowns on the additional grid points additional equations are provided with the requirement that the submatrix (the *interface coupling matrix*)

$$C \equiv \begin{bmatrix} C_{\ell\ell} & C_{\ell r} \\ C_{r\ell} & C_{rr} \end{bmatrix} \tag{3}$$

is nonsingular as for nonsingular C the solution $\mathbf{y}_{\approx}$ of (2) is unique, $\tilde{y}_\ell = y_\ell$ and $\tilde{y}_r = y_r$, and the restriction of $\mathbf{y}_{\approx}$ to $\mathbf{y}$ is the unique solution of the original linear system (1) ([9, Theorem 1], [8, Theorem 1.2.1]).
2. Splits the matrix $\mathbf{B}_C = \mathbf{M}_C - \mathbf{N}_C$ in a part $\mathbf{M}_C$, the boxed parts in (2) that do not map elements from one subgrid to the other subgrid and a remaining part $\mathbf{N}_C$ that couples the subgrids via the discretized interface with a relatively small number of nonzero elements. (Therefore matrix vector multiplication with $\mathbf{B}_C$ can be implemented efficiently on distributed memory computers.)
3. Tunes the interface coupling matrix C defined in (3) such that error components due to domain decomposition are damped in the Richardson iteration

$$\mathbf{y}_{\approx}^{(i+1)} = \mathbf{y}_{\approx}^{(i)} + \mathbf{M}_C^1\,(\mathbf{d}_0 - \mathbf{B}_C\,\mathbf{y}_{\approx}^{(i)}). \tag{4}$$

Note $\mathbf{M}_C^{-1}\mathbf{B}_C = \mathbf{I} - \mathbf{M}_C^{-1}\mathbf{N}_C$, therefore error components are propagated by $\mathbf{M}_C^{-1}\mathbf{N}_C$.
4. Computes a solution of the enhanced linear system from (4) or with a more general Krylov method like GMRES [6] with $\mathscr{K}_m(\mathbf{M}_C^{-1}\,\mathbf{B}_C, \mathbf{M}_C^{-1}\,\mathbf{d}_0) \equiv$
$\mathrm{span}(\mathbf{M}_C^{-1}\,\mathbf{d}_0, \mathbf{M}_C^{-1}\,\mathbf{B}_C\,\mathbf{M}_C^{-1}\,\mathbf{d}_0, \ldots, (\mathbf{M}_C^{-1}\,\mathbf{B}_C)^{m-1}\,\mathbf{M}_C^{-1}\,\mathbf{d}_0)$.

The key idea is to use the degrees of freedom, that we have created by the introduction of additional unknowns near the interface, for damping the error components. For this purpose, the spectral properties of $\mathbf{M}_C^{-1}\mathbf{N}_C$ for the specific underlying PDE are analyzed. With results of this analysis, optimal coupling parameters can be estimated, i.e. the interface coupling matrix C defined in (3) can be tuned. In this way error components due to the splitting are damped "as much as possible", optimal choices result in a coupling that annihilates the outflow from one domain to another: absorbing boundary conditions. This leads effectively to almost uncoupled subproblems at subdomains. As a consequence, the number of iterations required for convergence is minimal with minimal communication overhead (due to the explicit step with $\mathbf{N}_C$) between subdomains: an ideal situation for implementation on parallel computers and/or distributed memory.

3 Jacobi-Davidson

For a standard eigenvalue problem $\mathbf{A}\,\mathbf{x} = \lambda\,\mathbf{x}$ each iteration Jacobi-Davidson [7]
1. Extracts an approximate eigenpair $(\theta,\mathbf{u}) \approx (\lambda,\mathbf{x})$ from a search subspace $\mathbf{V}$: construct $H \equiv \mathbf{V}^*\mathbf{A}\mathbf{V}$, solve $H\,s = \theta\,s$, compute $\mathbf{u} = \mathbf{V}\,s$.
2. Corrects the approximate eigenvector $\mathbf{u}$ with a correction $\mathbf{t} \perp \mathbf{u}$ that is computed from the *correction* equation:

$$\mathbf{P}\mathbf{B}\mathbf{P}\,\mathbf{t} = \mathbf{r} \quad \text{where} \quad \mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u}\,\mathbf{u}^*}{\mathbf{u}^*\mathbf{u}}, \mathbf{B} \equiv \mathbf{A} - \theta\,\mathbf{I}, \quad \text{and} \quad \mathbf{r} \equiv -\mathbf{B}\,\mathbf{u}. \tag{5}$$

3. Expands the search subspace with the correction $\mathbf{t}$: $\mathbf{V}_{new} = [\mathbf{V} \mid \mathbf{t}^\perp]$ where $\mathbf{t}^\perp \perp \mathbf{V}$.

The linear system described by the correction equation (5) may be highly indefinite and is given in an unusual manner so that the application of the domain decomposition technique needed further development and special attention.

Similar to the enhancements (1) of the linear system (2) in § 2, the following components of the correction equation are enhanced: the matrix $\mathbf{B} \equiv \mathbf{A} - \theta\,\mathbf{I}$ to $\mathbf{B}_C$, the correction vector $\mathbf{t}$ to $\mathbf{t}_\approx$ and the vectors $\mathbf{u}$ and $\mathbf{r}$ to $\mathbf{u}_0$ and $\mathbf{r}_0$. With these enhancements, a correction $\mathbf{t}_\approx \perp \mathbf{u}_0$ is computed from the following enhanced correction equation [3, §3.3.2]:

$$\mathbf{P}\mathbf{B}_C\mathbf{P}\,\mathbf{t}_\approx = \mathbf{r}_0 \quad \text{with} \quad \mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u}_0\mathbf{u}_0^*}{\mathbf{u}_0^*\mathbf{u}_0}. \tag{6}$$

The preconditioner $\mathbf{M}_C$ for $\mathbf{B}_C$ is constructed in the same way as the ordinary linear system case shown by the boxed parts in (2). However, because of the indefiniteness, for the correction equation the matrices $\mathbf{B}_C$ and $\mathbf{M}_C$ are accompanied by projections. Both for left and right preconditioning the projection is as follows:

$$\mathbf{P}' \equiv \mathbf{I} - \frac{\mathbf{M}_C^{-1}\mathbf{u}_0\,\mathbf{u}_0^*}{\mathbf{u}_0^*\mathbf{M}_C^{-1}\mathbf{u}_0}. \tag{7}$$

In case of left preconditioning (for right preconditioning see [3, §3.3.3]) we compute approximate solutions to the correction equation from

$$\mathbf{P}'\mathbf{M}_C^{-1}\mathbf{B}_C\mathbf{P}'\mathbf{t}_{\approx} = \mathbf{P}'\mathbf{M}_C^{-1}\mathbf{r}_0. \tag{8}$$

However, there is more to gain. For approximate solutions of the correction equation with a preconditioned Krylov method, the Jacobi-Davidson method is an accelerated inexact Newton method that consists of two nested iterative solvers. In the innerloop of Jacobi-Davidson a search subspace for the (approximate) solution of the correction equation is built up by powers of $\mathbf{M}_C^{-1}(\mathbf{A}-\theta\mathbf{I})$ for fixed θ. In the outerloop a search subspace for the (approximate) solution of the eigenvalue problem is built up by powers of $\mathbf{M}_C^{-1}(\mathbf{A}-\theta\mathbf{I})$ for variable θ. As θ varies slightly in succeeding outer iterations, one may take advantage of the nesting by applying the domain decomposition technique to the outer loop as was the subject of [4]. This effectively leaded to two different processes:

- Jacobi-Davidson with *enhanced inner loop*, enhancement at intermediate level with enhanced correction equation (6) and
- Jacobi-Davidson with *enhanced outer loop*, enhancement at highest level with a slightly different correction equation

$$\mathbf{P}\mathbf{B}_C\mathbf{P}\mathbf{t}_{\approx} = \mathbf{r}_{\approx} \quad \text{with} \quad \mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u}_0\mathbf{u}_0^*}{\mathbf{u}_0^*\mathbf{u}_0}. \tag{9}$$

The amount of work for both processes per outer iteration is almost the same. However, Jacobi-Davidson with enhanced outer loop turned out to be faster as it damps remaining error components from the previous outer iteration in the next one.

4 Scaling experiments

For the two processes, in [4, § 5.1] different eigenvalue problems have been considered including variable coefficients and large jumps. Here, to investigate practical aspects for parallel performance, we consider the eigenvalue problem for the Laplace operator as results for different numbers of subdomains show more regular behavior (see for instance Fig. 3 in [4]). Except for the first experiment about different decompositions, in all experiments we take for the domain Ω the unit square, decompose Ω in p square subdomains, and cover each subdomain by a 256×256 subgrid. Jacobi-Davidson is started with a parabolic shaped vector $x(1-x)y(1-y)$ for $0 \le x \le 1$ and $0 \le y \le 1$ (see also [3, § 3.5.1]) to compute the most global eigenvector (for which the corresponding eigenvalue is the closest one to zero) of the two-dimensional Laplace operator on Ω until the residual norm of the approximate eigenpair is less than 10^{-9}. We apply right preconditioning in the enhanced correction equation for exact solves with the preconditioner (i.e. exact subdomain solves) to enable a Schur complement approach. The preconditioner $\mathbf{M}_C$ is constructed only once, at the first Jacobi-Davidson outer iteration. The remaining linear system is solved with GMRES [6].

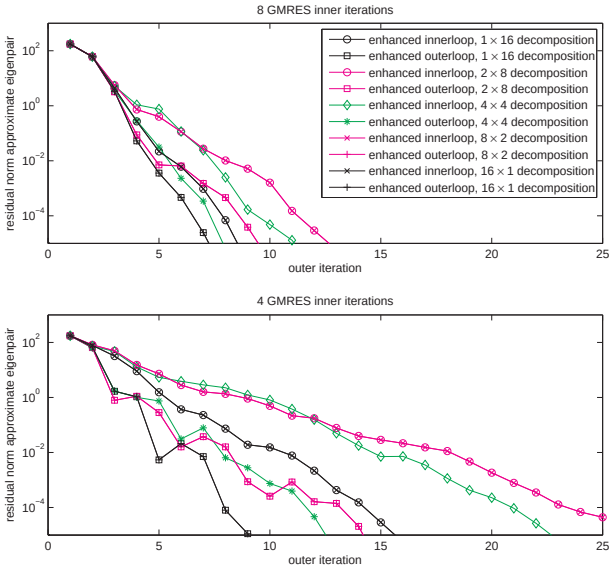


Fig. 2 Residual norm of the approximate eigenpair as a function of the Jacobi-Davidson outer iteration for the different decompositions with GMRES(8) (top) and GMRES(4) (bottom).

Implementation is in Fortran77 with calls to BLAS, LAPACK, and MPI. Note, however, that Fortran compiler, BLAS, LAPACK, and MPI versions differ on the specific hardware which is of influence on the (parallel) performance. Scaling experiments are performed on the following hardware:

- Curie linux-cluster (2 Intel eight 2.7 GHz core E5-2680 node, InfiniBand QDR, Intel Fortran 12, BLAS/LAPACK from MKL, Bull X MPI),
- H4+ linux-cluster (1 Intel quad 3.4 GHz core i7-2600 node, 1 GB/s Gigabit Ethernet, Intel Fortran 11, MPICH2),
- IBM POWER5+ system Huygens (16 IBM single 1.9 GHz core Power5+ node, 1.2 GB/s InfiniBand, XL Fortran 10, BLAS from ESSL, MPI from IBM PE),
- IBM POWER6 system Huygens (16 IBM dual 4.7 GHz core Power6 node, 160 GB/s InfiniBand, XL Fortran 12, BLAS from ESSL, MPI from IBM PE),
- Lisa 2008 linux-cluster (1 Intel Xeon 3.4E GHz core node, 800 MB/s InfiniBand, GFortran, MPICH2),
- Lisa 2012 linux-cluster (2 Intel eight 1.8 GHz core Xeon E5-2650L node, Intel Fortran 12, BLAS/LAPACK from MKL, OpenMPI),

On the H4+ and Lisa 2008 linux-clusters one subdomain is assigned to one node. On the other hardware one subdomain is assigned to one core. Results presented here are averages of three measured wall-clock times.

First we study different decompositions for a fixed number of subdomains for the same (discretized) eigenvalue problem. We keep the overall grid fixed to a size of 1024×1024 gridpoints and consider configurations with a 1×16, 2×8, 4×4, 8

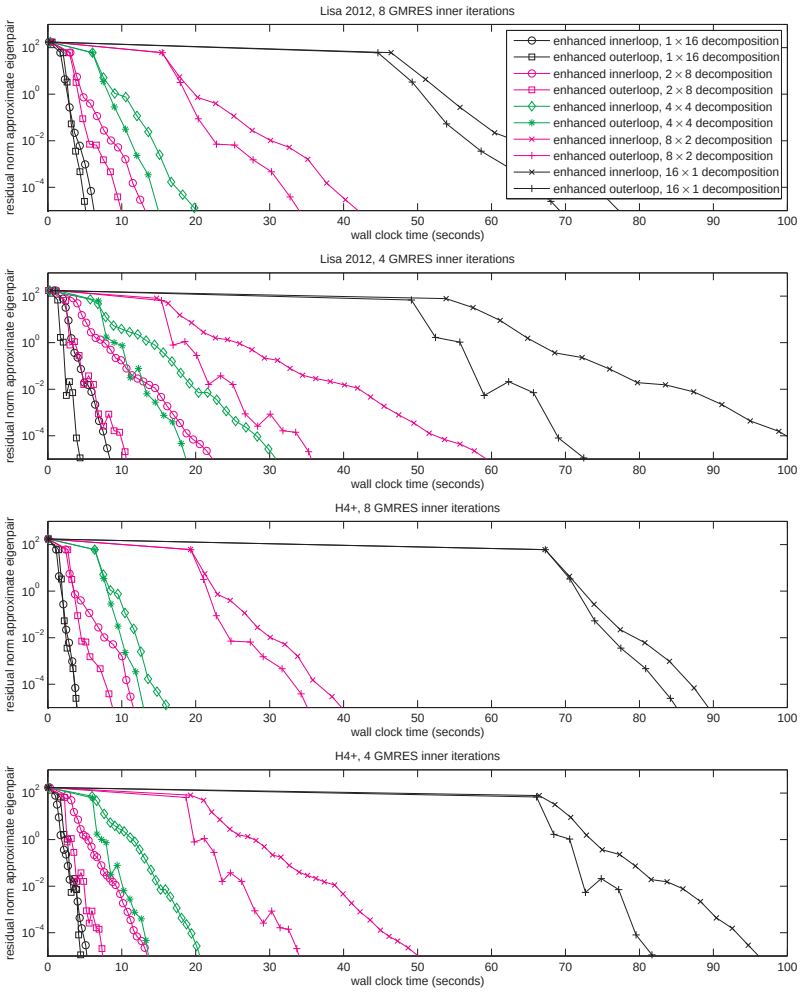


Fig. 3 Residual norm of the approximate eigenpair as a function of the wall clock time for the different decompositions. Shown are both enhanced innerloop and enhanced outerloop for the Lisa 2012 and H4+ linux-cluster and a fixed number of 8 and 4 inner iterations with GMRES.

$\times$ 2, and 16 $\times$ 1 decomposition, respectively (resulting in subgrids of size 1024 $\times$ 64, 512 $\times$ 128, 256 $\times$ 256, 128 $\times$ 512, and 64 $\times$ 1024, respectively). So the number of subdomains is 16 with 65536 unknowns per subdomain in all configurations, but the subdomains differ in shape. Fig. 2 shows the residual norm of the approximate eigenpair as a function of the Jacobi-Davidson outer iteration for the different decompositions. Shown are both enhanced innerloop and enhanced outerloop for a fixed number of 8 (top) and 4 (bottom) inner iterations with GMRES. As expected, the convergence histories for configurations which are mirrored (for instance 2 $\times$ 8 and 8 $\times$ 2) coincide. Decomposition in only one direction needs the least number

of outer iterations for convergence. For the tuning of the coupling between the subdomains we only took into consideration the one dimensional character of the error modes. For decompositions in two directions error modes will have a two dimensional character and are therefore harder to damp. Fig. 3 shows the residual norm of the approximate eigenpair as a function of wall clock time for the different decompositions. Shown are both enhanced innerloop and enhanced outerloop for the Lisa 2012 and H4+ linux-cluster and a fixed number of 8 and 4 inner iterations with GMRES. By comparing the mirrored configurations it can be observed that the grid ordering may significantly lower the performance. This is mainly in the construction of the preconditioner with LAPACK (initial horizontal lines in the figure). Although processors of the H4+ linux-cluster are faster, use of the MKL implementation of LAPACK resulted in a faster construction of the preconditioner at the Lisa 2012 linux-cluster. After the construction of the preconditioner, the process at the H4+ linux-cluster goes faster than the Lisa 2012 linux-cluster. At the H4+ linux-cluster communication is between 16 nodes over a relatively slow network, at the Lisa 2012 linux-cluster communication is fast inside a 16 core node with shared memory. So, we may conclude that the process is dominated by computational work. This confirms the remarks at the end of § 2 about the minimal communication overhead.

For the massively parallel behavior, we first extend Fig. 6 from [4] with results from (weak) scaling experiments on more recent hardware (IBM POWER6 system Huygens, Curie, and H4+). In Fig. 4 it can be observed that the trend holds, but now for lower wall clock times as processor speed has increased further for the more recent hardware.

To further investigate the weak scaling we start with a decomposition in 16 subdomains (on 1 node with 16 cores) on the Curie linux-cluster and increase everytime the number of subdomains in both directions with a factor 2. From 16, 64, 256, 1024, 4096 to 16384 subdomains (cores), resulting in up to more than 10^9 unknowns. For an efficient overall method, we will now use (see [1, §4])

$$\|\mathbf{r}^{(i)}\|_2 < 2^{-j} \|\mathbf{r}^{(0)}\|_2 \tag{10}$$

as a stopping criterion for the inner iterations (GMRES) at the j^{th} Jacobi-Davidson outer iteration. Here $\mathbf{r}^{(0)}$ is the residual at the start of the inner iterations and $\mathbf{r}^{(i)}$ the residual at the i^{th} inner iteration. Fig. 5 shows the results for Jacobi-Davidson with enhanced outerloop. Note that in this figure we choose the scaling of the x-axis to be quadratic to have a better impression. The figure indicates that for a large number of subdomains the wall clock doubles when the number of subdomains increases in both directions with a factor 2. This can be explained from the local behavior of the error modes due to domain decomposition: mainly one dimensional near the interface. The additional work to damp these error modes effectively depends on this local behavior.

Acknowledgements We thank SURFsara Computing and Networking Services (www.surfsara.nl) for their support in using the Power5+/6 system Huygens and Lisa linux-cluster. We acknowledge that the results in this paper have been achieved using the PRACE Research Infrastructure resource Curie based in France at CEA.

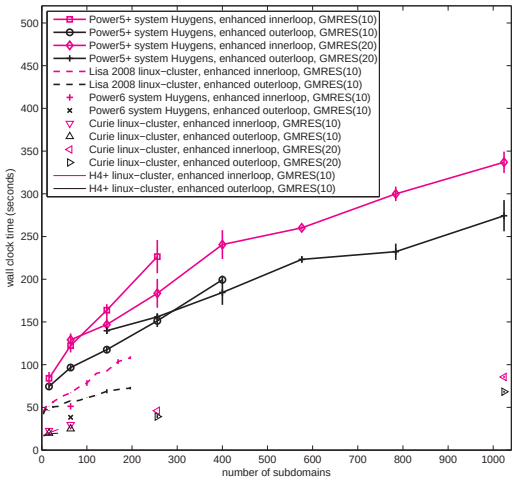


Fig. 4 Massively parallel behavior on different hardware.

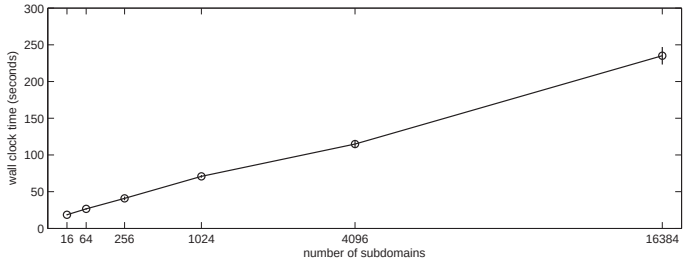


Fig. 5 Massively parallel behavior on the Curie linux-cluster (quadratic scaling of the x-axis).

References

1. Fokkema, D.R., Sleijpen, G.L.G., van der Vorst, H.A.: Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. SIAM J. Sci. Comput. **20**, 94–125 (1998)
2. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**, 699–731 (2006)
3. Genseberger, M.: Domain decomposition in the Jacobi-Davidson method for eigenproblems. Ph.D. thesis, Utrecht University (2001)
4. Genseberger, M.: Improving the parallel performance of a domain decomposition preconditioning technique in the Jacobi-Davidson method for large scale eigenvalue problems. Applied Numerical Mathematics **60**, 1083–1099 (2010)
5. Genseberger, M., Sleijpen, G.L.G., van der Vorst, H.A.: An optimized Schwarz method in the Jacobi-Davidson method for eigenvalue problems. Domain Decomposition Methods in Science and Engineering, pp. 289–296. UNAM (2003)
6. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comp. **7**, 856–869 (1986)
7. Sleijpen, G.L.G., van der Vorst, H.A.: A Jacobi-Davidson iteration method for linear eigenvalue problems. SIAM J. Matrix Anal. Appl. **17**, 401–425 (1996)
8. Tan, K.H.: Local coupling in domain decomposition. Ph.D. thesis, Utrecht University (1995)
9. Tan, K.H., Borsboom, M.J.A.: On generalized Schwarz coupling applied to advection- dominated problems. Domain Decomposition Methods in Scientific and Engineering Computing, pp. 125–130. Amer. Math. Soc. (1994)

Low-Rank Update of the Restricted Additive Schwarz Preconditioner for Nonlinear Systems

Laurent Berenguer[1] and Damien Tromeur-Dervout[1]

We consider the solution of differential equations of the form Eq.(1) for a given initial condition $y(0) = y_0$ and suitable boundary conditions.

$$M\dot{y} = g(y,t) \tag{1}$$

In Equation (1), $g \in C^1(\Omega, \mathbb{R}^n)$, for Ω an open set in $\mathbb{R}^n \times \mathbb{R}^*$ and $M \in \mathbb{R}^{n\times n}$. This equation is called a linear differential-algebraic equation (DAE) if the matrix M is singular. The time discretization of Eq. (1) via backward differentiation formulas leads to solving a system of nonlinear equations $f(y) = 0$ for $f : \mathbb{R}^n \to \mathbb{R}^n$ at each time step. These equations are generally solved by Newton-like methods which require the solution of numerous linear systems of the form:

$$J_k \Delta x_k = -f(x_k) \tag{2}$$

where $J_k \in \mathbb{R}^{n\times n}$ is the Jacobian matrix of f at x_k, or an approximation of it. In this paper we deal with the solution of these linear systems by a parallel Krylov iterative method. The condition number of the matrix J_k can be very large, hence, a good preconditioner is required.

Preconditioners based on the additive Schwarz method are often used to precondition sparse linear systems. The combination of a Newton method with a Krylov method preconditioned by a Schwarz method is generally called Newton-Krylov-Schwarz [5] and has widely be applied to CFD problems (see for example [6, 14, 7]). In this paper we deal with the Restricted Additive Schwarz preconditioner [8]. Computing and solving such linear systems is generally the most time consuming part of ODE/DAE integration codes, even if there are usually only slight changes between two consecutive linear systems. When the analytic Jacobian matrix is not available, a finite difference scheme is commonly used to approximate it [12] or its matrix-vector product [15]. Another way to avoid the computation of the Jacobian matrix is to update it from one iteration to another using quasi-Newton methods [10] that converge superlinearly [4]. Since Krylov methods are used to solve Equation (2), providing a preconditioner is a critical point. A balance must be found between the ability of the preconditioner to reduce the number of Krylov iterations, and its computational cost. Then, one may want to update the preconditioner using the secant condition in order to improve its efficiency. This idea is not new, and has been widely discussed in [3, 2]. The aim of this paper is to extend these techniques to domain decomposition based preconditioners such as the Restricted Additive Schwarz

[1] Université de Lyon, Université Lyon 1, CNRS, UMR 5208 Institut Camille Jordan, 43 boulevard du 11 novembre 1918, 69622 Villeurbanne cedex, France, e-mail: {laurent.berenguer}{damien.tromeur-dervout}@univ-lyon1.fr

preconditioner. First, we present the Broyden update and its application to general preconditioners. Then we discuss the practical issues in applying this update to the RAS preconditioner. The third part is devoted to numerical experiments on the CFD problem of the lid-driven cavity.

1 The Update of the RAS Preconditioner

The preconditioned linear system of the Newton iterations can be written as:

$$G_k J(x_k)\Delta x_k = -G_k f(x_k) \tag{3}$$

or

$$J(x_k)G_k G_k^{-1}\Delta x_k = -f(x_k) \tag{4}$$

depending on which side the preconditioner is applied.

For the sake of simplicity, we use the notations $f_k = f(x_k)$ and $\Delta f_k = f_{k+1} - f_k$ in the following. The quasi-Newton update of G_k, that satisfies the secant condition:

$$\Delta x_k = G_{k+1}\Delta f_k \tag{5}$$

is given by:

$$G_{k+1} = G_k + (\Delta x_k - G_k \Delta f_k)\frac{v_k^T}{v^T \Delta f_k} \text{ for some } v_k \tag{6}$$

Usually, v_k is taken as Δf_k or $G_k^T \Delta x_k$:

- If $v_k = G_k^T \Delta x_k$, then G_{k+1} minimizes $\|G_{k+1}^{-1} - G_k^{-1}\|_F$.
- If $v_k = \Delta f_k$, then G_{k+1} minimizes $\|G_{k+1} - G_k\|_F$.

In both cases, the proof can be derived in straightforward manner from the proof of Theorem 4.1 in [10]. In general, it is not possible to give an estimation of the effect of the update of the preconditioner in terms of condition number. Nevertheless, it is possible to give a lower bound of condition number of the updated preconditioned linear system. Let $\{\sigma_k\}$ and $\{\tau_k\}$ be the singular values of $G_k J(x_{k+1})$ and $G_{k+1}J(x_{k+1}) = G_k J(x_{k+1}) + uw^T$ for $w^T = v^T J(x_{k+1})$. Then, the interlacing property of the singular values [13, Theorem 6.1] gives:

$$\begin{cases} \sigma_2 \leq \tau_1, \\ \sigma_{k+1} \leq \tau_k \leq \sigma_k,\ 1 < k < n \\ 0 \leq \tau_n \leq \sigma_{n-1}, \end{cases} \tag{7}$$

Then,

$$\kappa_2(G_{k+1}J(x_{k+1})) = \frac{\tau_1}{\tau_n} \geq \frac{\sigma_2}{\sigma_{n-1}}. \tag{8}$$

The same results can be derived for right preconditioned linear systems, since a rank-one update of the preconditioner linear system leads to a rank one modification of the preconditioned operator. This lower bound gives a limitation of the updating procedure: it will not be efficient if the preconditioned linear system has a large set of very high, or very low singular values.

Let us now illustrate the effect of the Broyden update on a manufactured problem. $-\Delta_{FD2}$ be the second order finite difference discrete 1D Laplacian operator for homogeneous boundary conditions, associated to the eigenpairs $\{(U_i,\lambda_i)\}_{1\leq i\leq n}$ such that $\lambda_i > \lambda_{i+1}$. We define the nonlinear function $F(v)$ vanishing for $v=0$ and its Jacobian $J(v)$ definite positive matrix, of eigenpairs $\{(\eta_i U_1 + U_i, \mu_i)\}_{1\leq i\leq n}$, with condition number $\kappa_2(J(v)) = \frac{\mu_1}{\mu_n}$:

$$F(v) \stackrel{def}{=} \underbrace{(v,v)U_1U_1^T v}_{nonlinear} - \underbrace{\Delta_{FD2} v}_{linear} \tag{9}$$

$$J(v)h \;=\; 2(v,h)U_1U_1^T v + (v,v)U_1U_1^T h - \Delta_{FD2} h \tag{10}$$

$$\mu_1 = (2(v,U_1)^2 + (v,v) + \lambda_1), \mu_i = \lambda_i, 2 \leq i \leq n, \tag{11}$$

$$\eta_1 = 0, \eta_i = \frac{2(v,U_1)(v,U_i)}{\mu_i - \mu_1}, 2 \leq i \leq n. \tag{12}$$

For the sake of simplicity in calculus, starting from $X^0 = x_1^0 U_1$ and $G_0 = J(X^0)^{-1}$, Newton's and Broyden's iterates give the same $X^1 = \frac{2(x_1^0)^3}{\mu_1^0} U_1$ and the eigenvalue of $G_1 J(X^1)$ associated to U_1 is given by:

$$(G_1 J(X^1)U_1 = \frac{{\lambda_1}^3 + 6\,(x_1^0)^2 {\lambda_1}^2 + 9\,\lambda_1 (x_1^0)^4 + 12\,(x_1^0)^6}{{\lambda_1}^3 + 7\,(x_1^0)^2 {\lambda_1}^2 + 17\,\lambda_1 (x_1^0)^4 + 19\,(x_1^0)^6} U_1$$

These results suggest that Z_1 is a good preconditioner for $J(X_1)$ if X^0 is close to the solution $X=0$, ($G_1 J(X^1)$ have the same $(n-1)$ eigenpairs $(U_i,1), 2 \leq i \leq n$).

2 Application to the RAS Preconditioner

The Restrictive Additive Schwarz preconditioner of the linear system $J(x)\Delta x = -f(x)$ decomposed in s overlapping subdomains, is given by:

$$M_{RAS}^{-1} = \sum_{i=1}^{s} \tilde{R}_i^T J^i(x)^{-1} R_i \tag{13}$$

where R_i is the restriction operator of the ith subdomain including the overlap, and $\tilde{R}_i$ is the restriction operator except that only interior nodes have a corresponding

nonzero line. The matrix $J^i(x)$ is the submatrix of $J(x)$ corresponding to the ith subdomain including the overlap. We propose to performing Broyden's updates starting from the RAS preconditioner $G_0 = M_{RAS}^{-1} = \sum_i \tilde{R}_i J^i(x)^{-1} R_i^T$.

Algorithm 1 gives an overview of the method for ($v_k = \Delta f_k$) within a time-stepper. Finding an optimal restarting criterion is out of the scope of this paper. One should notice that the restart may not happen at each time step. Hence, two simple strategies could be (1) to restart every r time steps, or (2) to restart when a maximum number of Krylov iterations has been reached for the solution of the previous linear system.

Algorithm 1 Time stepper with update of the RAS preconditioner

Require: restart parameter, initial guess x, $k = 0$
for each time step **do**
  // *Newton iterations:*
  repeat
    if restart **then**
      $G_0 \leftarrow \sum_i \tilde{R}_i^T J_i(x_0)^{-1} R_i$ // *Local LU factorizations*
      $k \leftarrow 0$
    end if
    solve $J(x)\Delta x = -f(x)$ with a Krylov method preconditioned by G_k.
    $x \leftarrow x + \Delta x$
    $G_{k+1} = G_k + (\Delta x_k - G_k \Delta f_k) \frac{f_k^T}{f_k^T \Delta f_k}$
    $k \leftarrow k+1$
  until convergence
end for

Therefore, even if G_0 is a sparse matrix, G_k is not. Consequently, the matrix G_k is never formed, we only compute its application to a vector. Let u_k be $(\Delta x_k - G_k \Delta f_k)/(v^T \Delta f_k)$ then the application of G_k to an arbitrary vector x depends on the choice made for v_k :

- For $v_k = \Delta f_k$ the application of the preconditioner can be rewritten as:

$$G_{k+1}x = G_0 x + \sum_{i=0}^{k} u_i v_i^T x = G_0 x + [u_0 \cdots u_k][v_0 \cdots v_k]^T x \tag{14}$$

  Hence, the additional cost of the application of G_k compared to G_0 is roughly two matrix-vector products of $n \times k$ matrices. Furthermore, the computation of u_k involves one application of G_k. One should also notice that the local LU factorizations can also be computed asynchronously, continuing Newton iterations during the computation of the restarted preconditioner.
- For $v_k = G_k^T \Delta x_k$, the explicit computation of v_k should be avoided because it involves G_k^T, so M_{RAS}^{-T} which cannot be easily computed. Then $G_{k+1}x$ is usually rewritten as in Eq. (15).

$$G_{k+1}x = \left(\prod_{i=k}^{0}(I - u_i \Delta x_i{}^T)\right) G_0 x \tag{15}$$

Following an idea of Martínez [16], Bergamaschi et al. proved in [3, theorem 3.6] that for G_0 and x_0 good enough initial guesses, the norm $\|I - G_k J(x_k)\|$ can be made arbitrarily small. Since the preconditioner is also reused from one time step to another, it slowly loses its efficiency and the algorithm must be restarted, which means recomputing G_0.

In terms of condition numbers, the preconditioner G_k is not expected to be more efficient than the RAS preconditioner M_{RAS}^{-1} of the current Jacobian matrix $J(x_k)$, but it's computational cost is less important: computing G_{k+1} form G_k does not involves LU factorizations unlike the computation of a new M_{RAS}^{-1}.

The efficiency of the updated preconditioner is expected to decrease from one time step to another, but this decrease should be slowed by the update. This decrease can be roughly explained by the fact that the convergence of Broyden's method is slower than the convergence of Newton's method. Thus, a restart of the algorithm is needed. This restart (Algo.1, step 4) consists in the computation of a new $G_0 = M_{RAS}^{-1}$ (i.e. new local LU factorizations).

One of the main drawbacks of the method presented here is the increase of the memory cost by two vectors per update. A few techniques can be used to reduce this memory cost: the simplest one consists in restarting the algorithm when a maximum number of updates is reached. One may also compress the updates using a truncated SVD of $[u_0 \cdots u_k][v_0 \cdots v_k]^T$ [18].

The parallelism of Equations (14) and (15) should also be discussed:

- The application of the preconditioner in Eq. (14) to a vector x involves global communications since the matrices $[u_0 \cdots u_k]$ and $[v_0 \cdots v_k]$ are dense, and distributed over the processors. Then, depending on the implementation, Eq. (14) requires an additional global reduction of k values, or k reductions where the $k-1$ first are overlapped by computations.
- The parallel implementation of Eq. (15) requires k sequential collective reductions. Hence, one should not use $v_k = G_k^T \Delta x_k$ for a parallel implementation on distributed memory computers.

3 Numerical experiments

Let us first give a numerical illustration of the model problem $F(v) = c$ where F is from Eq. (9), $c \in \mathbb{R}^{100}$ an arbitrary vector, and starting from $G_0 = M_{RAS}^{-1}(-\Delta_{FD2})$. Then, the condition numbers are: $\kappa_2(J(X^1)) = 1.8 \times 10^9$, $\kappa_2(G_0 J(X^1)) = 1.7 \times 10^8$ and $\kappa_2(G_1 J(X^1)) = 1.2 \times 10^3$ when the preconditioner G_0 is updated with Broyden's update. This suggests that the update of the preconditioner has efficiently reduced the effect of the first eigenvalue of $J(X^1)$. We now consider the lid-driven cavity problem on the unit square. The PETSc library [1] was used for the implementa-

tion. In particular, the implementation of the following (u,v,ω,T)-formulation is provided as a PETSc example [9]. The linear solver used in these experiments is a BiCGstab [17] and Jacobian matrices are approximated by a coloring method.

$$\begin{cases} -\Delta(u) - \nabla_y(\omega) = 0 \\ -\Delta(v) + \nabla_x(\omega) = 0 \\ \dot{\omega} - \Delta(\omega) + \nabla \cdot ([u \times \omega, v \times \omega]) - \nabla_x(T) = 0 \\ \dot{T} - \Delta(T) + \nabla \cdot ([u \times T, v \times T]) = 0 \end{cases} \tag{16}$$

Where u and v are the two components of the velocity field, $\omega = -\nabla_y u + \nabla_x v$ is the vorticity and T the temperature. The space discretization is performed on a regular grid with a five-point stencil and the time discretization is a backward Euler scheme. The lid-velocity $u(x,0)$ is a nonzero constant, the other boundary conditions satisfy $u = v = 0$, $T = 0$ on the left wall, and $T = 1$ on the right wall, $\partial T/\partial y = 0$ on the top and the bottom. A fixed time step length has been used for the simulation, excepted for the very first time steps. The initial solution is zero everywhere excepted on the walls, and the solution at the previous time step is used as the initial guess for the current time step. In the following results, the linear systems are right preconditioned and G_0 is the RAS preconditioner of the current approximation of the Jacobian matrix, and the overlapping size is one. The reason is that when the left preconditioning technique is used, the natural stopping criterion of the Kyrlov method is based on the norm of the preconditioned residual. Hence, in order to compare two different preconditioners, one should use a stopping criterion based on the norm of true residual. The Newton iterations are stopped (i.e. the time step is accepted) when the absolute norm of the residual is lower than 10^{-6}.

Table 1 Comparison of the updated and the frozen preconditioner for a 512×512 grid decomposed in 8×8 subdomains. The lid velocity is $u(x,0) = 500$ and the time step length is 10^{-3}. 1000 time steps are performed, and the sum of all the BiCGstab iterations is given. The algorithm is restarted every f_r time steps, and the walltimes are given in seconds.

	With update		Without update		Saved	Saved
f_r	BiCGstab it.	Walltime	BiCGstab it.	Walltime	iterations (%)	walltime (%)
1	34483	4729	34882	4744	1.144	0.309
5	34572	3820	35230	3850	1.868	0.779
40	35165	3609	35946	3649	2.173	1.085
60	35785	3619	36249	3625	1.280	0.159
80	36110	3653	36693	3670	1.589	0.461

Table 1 compares the total number of BiCGstab iterations with and without the rank-one update, for different frequencies of restarting. A frequency of restarting f_r of 10 means that 100 local LU factorizations have been computed on each processors during the 1000 time steps. There is actually between one and three Newton iterations per time steps. This results show that the total number of Krylov iterations is slightly reduced by the updating method. If we take into account only the 580

time steps for which three Newton iterations have been performed, then 3.79% of the Krylov iterations have been saved. the additional cost of the application of the preconditioner is the reason why the proportion of Krylov iterations that are saved is greater than the proportion of saved computational time.

Table 2 Number BiCGstab iterations for the updated and the frozen preconditioner. The grid decomposition is regular, using the same number of subdomains in each direction. 1000 time steps of length 10^{-3} are performed. The algorithm is restarted every 40 time steps.

Processors	Grid size	Lid velocity	With update	Without update	Saved .it (%)
8	128^2	100	9009	9474	4.908
8	128^2	300	16358	16748	2.329
16	128^2	100	12724	13275	4.150
16	128^2	300	17961	18345	2.093
16	256^2	300	20011	20805	3.816
64	256^2	300	28408	30114	5.665
64	256^2	500	32599	32889	0.882

Table 2 compares the number of BiCGstab iterations for different sizes of grid and lid velocities. This results show that the Broyden update of the preconditioner may leads to a significant reduction of the number of Krylov iterations. For a restarting frequency of 40, the percentage of saved iterations generally decreases when the lid velocity is increased. This suggests that a more appropriate restarting algorithm should be designed in order to preserve the efficiency of the update. It should be noticed that the results presented above are obtained for a fixed time step length. The efficiency of the update is expected to change if an adaptive time stepping algorithm is used since the step length is present on the diagonal of the Jacobian matrix.

4 Conclusions

We presented a very simple procedure to update the RAS preconditioner without loss of parallelism. This update leads to a decrease of the number of Krylov iterations, especially for the time steps that requires the largest number of Newton iterations. However, further developments are needed to achieve an efficient method. This quasi-Newton update of the preconditioner should be used with a well-parametrized restarting procedure, since the efficiency of the preconditioner decreases from one iteration to another. A natural extension of this work is to use higher-rank updates, like the multisecant update [11]. Techniques such as partial updates, or relaxed updates should also be investigated since they are expected to significantly improve the numerical efficiency of the updated preconditioner.

Acknowledgements This work has been supported by the French National Agency of Research (project ANR-12-MONU-0012 H2MNO4). Laurent Berenguer held a doctoral fellowship (32116

Euros in 2012-2013) from the Région Rhône-Alpes. Authors also thank the Center for the Development of Parallel Scientific Computing (CDCSP) of the University of Lyon 1 for providing us with computing resources.

References

1. Balay, S., Brown, J., , Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.3, Argonne National Laboratory (2012)
2. Bergamaschi, L., Bru, R., Martínez, A.: Low-rank update of preconditioners for the inexact Newton method with SPD Jacobian. Math. Comput. Modelling **54**(7-8), 1863–1873 (2011)
3. Bergamaschi, L., Bru, R., Martínez, A., Putti, M.: Quasi-Newton preconditioners for the inexact Newton method. Electron. Trans. Numer. Anal. **23**, 76–87 (electronic) (2006)
4. Broyden, C.G., Dennis Jr., J.E., Moré, J.J.: On the local and superlinear convergence of quasi-Newton methods. J. Inst. Math. Appl. **12**, 223–245 (1973)
5. Cai, X.C., Gropp, W.D., Keyes, D.E., D., T.M.: Newton-KrylovSchwarz methods in CFD. In: Proceedings of the International Workshop on Numerical Methods for the NavierStokes Equations,, pp. 17–30. Vieweg, Braunschweig (1995)
6. Cai, X.C., Keyes, D.E., Venkatakrishnan, V.: Newton-Krylov-Schwarz: An implicit solver for CFD. In: Proceedings of the Eighth International Conference on Domain Decomposition Methods, pp. 387–400. Wiley, New York (1997)
7. Cai, X.C., Keyes, D.E., Young, D.P.: A nonlinear additive Schwarz preconditioned inexact Newton method for shocked duct flows. In: Domain decomposition methods in science and engineering (Lyon, 2000), Theory Eng. Appl. Comput. Methods, pp. 345–352. Internat. Center Numer. Methods Eng. (CIMNE), Barcelona (2002)
8. Cai, X.C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J. Sci. Comput. **21**(2), 792–797 (electronic) (1999)
9. Coffey, T.S., Kelley, C.T., Keyes, D.E.: Pseudotransient continuation and differential-algebraic equations. SIAM J. Sci. Comput. **25**(2), 553–569 (2003)
10. Dennis Jr., J.E., Moré, J.J.: Quasi-Newton methods, motivation and theory. SIAM Rev. **19**(1), 46–89 (1977)
11. Fang, H.r., Saad, Y.: Two classes of multisecant methods for nonlinear acceleration. Numer. Linear Algebra Appl. **16**(3), 197–221 (2009)
12. Gebremedhin, A.H., Manne, F., Pothen, A.: What color is your Jacobian? Graph coloring for computing derivatives. SIAM Rev. **47**(4), 629–705 (electronic) (2005)
13. Greif, C., Varah, J.M.: Minimizing the condition number for small rank modifications. SIAM J. Matrix Anal. Appl. **29**(1), 82–97 (electronic) (2006/07)
14. Keyes, D.E.: Aerodynamic applications of Newton-Krylov-Schwarz solvers. In: Fourteenth International Conference on Numerical Methods in Fluid Dynamics (Bangalore, 1994), *Lecture Notes in Phys.*, vol. 453, pp. 1–20. Springer, Berlin (1995)
15. Knoll, D.A., Keyes, D.E.: Jacobian-free Newton-Krylov methods: a survey of approaches and applications. J. Comput. Phys. **193**(2), 357–397 (2004)
16. Martínez, J.M.: An extension of the theory of secant preconditioners. J. Comput. Appl. Math. **60**(1-2), 115–125 (1995). Linear/nonlinear iterative methods and verification of solution (Matsuyama, 1993)
17. van der Vorst, H.A.: Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **13**(2), 631–644 (1992)
18. Ziani, M., Guyomarc'h, F.: An autoadaptative limited memory Broyden's method to solve systems of nonlinear equations. Appl. Math. Comput. **205**(1), 202–211 (2008)

GMRES acceleration of restricted Schwarz iterations

Pacull[1] and Aubert[1]

We present here an analysis of the Richardson iterations preconditioned by either the restricted additive [2] or multiplicative Schwarz [6] operators, and the associated GMRES Krylov sub-space acceleration. The framework of study is purely algebraic and general sparse unsymmetrical and indefinite matrices are considered. This paper can be seen as an extension of [1, 10], in which a block preconditioned system is downsized to an interface system. The following study is circumscribed to restricted Schwarz preconditioners.

At first, the equivalence between the primary and interface iterations is described. Then, the interface system operator is depicted as a Schur complement of the permuted preconditioned global matrix. Finally, the benefit of the Krylov sub-space acceleration of the interface iterations, over the primary ones, is exhibited. Note that exact solves of the sub-domain problems is assumed throughout.

The linear system to solve is :

$$Au = f \tag{1}$$

with $A \in \mathbb{R}^{n \times n}$, $u \in \mathbb{R}^n$ and $f \in \mathbb{R}^n$. We assume that A is close to structurally symmetric, which is a common property of matrices originating from PDE problems.

As a preparatory step, we start by introducing the vertex-based partitioning process and the notations used hereafter.

1 Introduction

1.1 Graph partitioning and overlap

We denote $\mathscr{G}$ the adjacency graph of matrix A, $\mathscr{V} = \{1, 2, ..., n\}$ the nodes of $\mathscr{G}$, and $\mathscr{E}$ the edges, which correspond to the non-zero off-diagonal elements of A. The graph $\mathscr{G}$ is considered to be undirected: given an unordered pair of distinct nodes $(v_1, v_2) \in \mathscr{V}^2$, we have $(v_1, v_2) \in \mathscr{E}$ if and only if $A(v_1, v_2) \neq 0$ or $A(v_2, v_1) \neq 0$.

A non-overlapping partition of $\mathscr{V}$ with p sub-domains corresponds to p non-empty sub-sets, $\{\mathscr{V}_i\}_{1 \leq i \leq p}$, such that $\mathscr{V} = \cup_{i=1}^{p} \mathscr{V}_i$ and $\mathscr{V}_j \cap \mathscr{V}_k = \varnothing$ for $1 \leq j < k \leq p$. The usual goal when performing this graph-partitioning task is to minimize the overall edge cut, which is the total number of edges $(v_i, v_j) \in \mathscr{E}$ with v_i and v_j belonging to distinct sub-domains, while equilibrating the number of nodes per sub-domain to approximatively n/p. Dealing with p equal sub-sets aims at balancing the dis-

[1] Fluorem, 64 Chemin des Mouilles 69130 Ecully, France e-mail: {fpacull}{saubert}@fluorem.com

tributed computational and memory load per processor. Minimizing the number of edges crossing the partition boundaries results in a reduced communication volume between processors.

Increasing the δ-overlap is beneficial regarding the convergence rate of Schwarz methods (see [6] for example): starting from $\mathscr{V}_{i,0} \equiv \mathscr{V}_i$, this consists in growing recursively each sub-set $\mathscr{V}_{i,\delta}$ by adding some of the adjacent nodes, in order to form a larger set $\mathscr{V}_{i,\delta+1}$.

For each sub-domain and for each δ level, $n_{i,\delta} \equiv |\mathscr{V}_{i,\delta}|$ refers to the cardinality of the node sub-set.

1.2 Notations regarding restrictions operators

Similarly to what is done in [9], three different sub-sets of nodes are defined in association with a given sub-domain $\mathscr{V}_{i,\delta}$: $\mathscr{V}_{i,\delta}^{int}$, $\mathscr{V}_{i,\delta}^{loc}$ and $\mathscr{V}_{i,\delta}^{ext}$. The internal nodes $\mathscr{V}_{i,\delta}^{int}$ are the nodes of $\mathscr{V}_{i,\delta}$ that have their graph neighborhood fully included in $\mathscr{V}_{i,\delta}$. The local interface nodes $\mathscr{V}_{i,\delta}^{loc}$ are the nodes of $\mathscr{V}_{i,\delta}$ that have a least one of their neighbors outside of $\mathscr{V}_{i,\delta}$. Finally, the external interface nodes $\mathscr{V}_{i,\delta}^{ext}$ are the nodes that do not belong to $\mathscr{V}_{i,\delta}$, but which have at least one of their neighbors within $\mathscr{V}_{i,\delta}$.

Note that $\mathscr{V}_{i,\delta}^{ext}$ is the set of candidate nodes for growing the sub-set $\mathscr{V}_{i,\delta}$: $\mathscr{V}_{i,\delta+1} \subseteq \mathscr{V}_{i,\delta} \cup \mathscr{V}_{i,\delta}^{ext}$.

An important sub-set of nodes for our study is the global set of external interface node, simply called the *interface nodes* hereafter: $\mathscr{V}_{\delta}^{ext} \equiv \cup_{i=1}^{p} \mathscr{V}_{i,\delta}^{ext}$, with cardinality $n_{\delta}^{ext} \equiv |\mathscr{V}_{\delta}^{ext}|$. The complementary sub-set of $\mathscr{V}_{\delta}^{ext}$ is denoted by $\bar{\mathscr{V}}_{\delta}^{ext} \equiv \mathscr{V} \setminus \mathscr{V}_{\delta}^{ext}$.

In the following, notations from [7] are used to describe the different operators associated with the algebraic Schwarz preconditioners. For the i-th sub-domain, we denote $R_{i,\delta} \in \mathbb{R}^{n_{i,\delta} \times n}$ the restriction operator associated with $\mathscr{V}_{i,\delta}$. $R_{i,\delta}^{ext}$ is the restriction operator associated with $\mathscr{V}_{i,\delta}^{ext}$. The special restriction operator used in the restricted Schwarz iterations, is defined as follows: $\tilde{R}_{i,\delta} \equiv R_{i,\delta} R_{i,0}^T R_{i,0} \in \mathbb{R}^{n_{i,\delta} \times n}$.

The node sub-set $\bar{\mathscr{V}}_{i,\delta}$ refers to the following set difference: $\bar{\mathscr{V}}_{i,\delta} \equiv \mathscr{V} \setminus \mathscr{V}_{i,\delta}$, and $\bar{R}_{i,\delta}$ to the restriction operator associated with $\bar{\mathscr{V}}_{i,\delta}$. R_{δ}^{ext} and $\bar{R}_{\delta}^{ext}$ are the restriction operators associated with $\mathscr{V}_{\delta}^{ext}$ and $\bar{\mathscr{V}}_{\delta}^{ext}$ respectively.

The local parts of the operator A are the following ones: $A_{i,\delta} \equiv R_{i,\delta} A R_{i,\delta}^T$ for the inner coupling, and $A_{i,\delta}^{ext} \equiv R_{i,\delta} A R_{i,\delta}^{ext\,T}$ for the outer coupling.

Finally, the vector y stands for the vector of interface node unknowns

$$y = R_{\delta}^{ext} u \in \mathbb{R}^{n_{\delta}^{ext}} \tag{2}$$

while $x = \bar{R}_{\delta}^{ext} u \in \mathbb{R}^{n - n_{\delta}^{ext}}$ stands for the complementary unknowns, located at the non-interface nodes $\bar{\mathscr{V}}_{\delta}^{ext}$.

2 Richardson iterations with a restricted Schwarz preconditioner

The preconditioned Richardson iteration $u^{(k+1)} = u^{(k)} + M^{-1}(f - Au^{(k)})$, is expressed as the stationary iteration

$$u^{(k+1)} = F\,u^{(k)} + g \tag{3}$$

where $F = I - M^{-1}A$ and $g = M^{-1}f$ are the iteration matrix and vector. We only consider here the restricted additive (RAS) and multiplicative (RMS) Schwarz preconditioners, as defined for example in [6]:

$$F_{RAS,\delta} = I - \sum_{i=1}^{p} \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta} A \tag{4}$$

$$F_{RMS,\delta} = \prod_{i=p}^{1} \left(I - \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta} A \right) \tag{5}$$

As pointed out in [1, 10], under some specific conditions, the primary iteration (3) can be reduced to an equivalent interface iteration, in terms of the unknown y defined in (2):

$$y^{(k+1)} = Gy^{(k)} + h \tag{6}$$

In order to gain more insight into this interface system, let us derive the iteration (6) starting from (3). If the restriction R_δ^{ext} is applied to (3), we get the following iteration: $y^{(k+1)} = R_\delta^{ext} F x^{(k)} + h$, with $h \equiv R_\delta^{ext} g$. We now make use of the following relation:

$$\begin{aligned} R_{i,\delta}A &= R_{i,\delta}A(R_{i,\delta}^T R_{i,\delta} + \bar{R}_{i,\delta}^T \bar{R}_{i,\delta}) \\ &= A_{i,\delta}R_{i,\delta} + R_{i,\delta}A\bar{R}_{i,\delta}^T \bar{R}_{i,\delta} \\ &= A_{i,\delta}R_{i,\delta} + A_{i,\delta}^{ext} R_{i,\delta}^{ext} \end{aligned} \tag{7}$$

Thus, in the restricted additive Schwarz case, we have:

$$\begin{aligned} F_{RAS,\delta} &= I - \sum_{i=1}^{p} \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} (A_{i,\delta}R_{i,\delta} + A_{i,\delta}^{ext} R_{i,\delta}^{ext}) \\ &= I - \sum_{i=1}^{p} \tilde{R}_{i,\delta}^T R_{i,\delta} - \sum_{i=1}^{p} \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} A_{i,\delta}^{ext} R_{i,\delta}^{ext} \end{aligned} \tag{8}$$

Using the following equality, $\sum_{i=1}^{p} \tilde{R}_{i,\delta}^T R_{i,\delta} = \sum_{i=1}^{p} R_{i,0}^T R_{i,0} R_{i,\delta}^T R_{i,\delta} = \sum_{i=1}^{p} R_{i,0}^T R_{i,0} = I$, we get:

$$F_{RAS,\delta} = -\sum_{i=1}^{p} \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} A_{i,\delta}^{ext} R_{i,\delta}^{ext} \tag{9}$$

This shows that the iteration matrix $F_{RAS,\delta}$ only depends on the interface nodes.

For the multiplicative case, by using (7), we get:

$$\begin{aligned} F_{RMS,\delta} &= \prod_{i=p}^{1} \left(I - \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} R_{i,\delta} A \right) \\ &= \prod_{i=p}^{1} \left(I - R_{i,0}^T R_{i,0} - \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} A_{i,\delta}^{ext} R_{i,\delta}^{ext} \right) \end{aligned} \tag{10}$$

For simplicity reasons, we call a_i the left term in the parentheses and b_i the right term: $a_i \equiv I - R_{i,0}^T R_{i,0}$, $b_i \equiv \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} A_{i,\delta}^{ext} R_{i,\delta}^{ext}$. By noticing that $\prod_{i=p}^{1} a_i = 0$ and that $a_i b_j = b_j$ if $i \neq j$, we get:

$$F_{RMS,\delta} = \sum_{k=1}^{p} \sum_{p \geq i_1 > \ldots > i_k \geq 1} (-1)^k b_{i_1} \ldots b_{i_k} \tag{11}$$

The important thing is that the b_i terms only depend on the interface nodes, and so does $F_{RMS,\delta}$ consequently.

Hence we have, in both restricted Schwarz cases:

$$F = F R_\delta^{ext\,T} R_\delta^{ext} \text{ and } F \bar{R}_\delta^{ext\,T} \bar{R}_\delta^{ext} = 0 \tag{12}$$

Indeed we know from [10] that k belongs to $\bar{\mathcal{V}}_\delta^{ext}$ (that is, k is not an interface node) if and only if the k-th column of F is null, and if and only if the k-th column of M is equal to the k-th column of A.

We can now state that with the coherent initial interface conditions $y^{(0)} = R_\delta^{ext} u^{(0)}$, the following relation between $u^{(k)}$ and $y^{(k)}$ holds:

$$y^{(k+1)} = R_\delta^{ext} u^{(k+1)} = R_\delta^{ext} \left[F u^{(k)} + g \right] = G y^{(k)} + h \text{ for } k \geq 1 \tag{13}$$

The iteration matrix G can be expressed as follows: $G = R_\delta^{ext} F R_\delta^{ext\,T}$. Note that this relation holds whatever the initial condition $x^{(0)} = \bar{R}_\delta^{ext} u^{(0)}$ is.

We now focus on the interface system: $(I - G) y^{(\infty)} = h$.

3 Restricted Schwarz and Schur

In [3, 8], it is shown that a multiplicative Schwarz iterate is identical to a block Gauss-Seidel sweep applied to the Schur complement system on the interface unknowns, provided that coherent initial conditions are used. Similar results also holds between the additive Schwarz iterate and a block Jacobi sweep of the Schur complement system. The considered Schur complement S is related to the interface nodes of the non-overlapping partition. In the overlapping case, it is possible to decom-

pose the sub-domains into smaller disjoint parts and express the global matrix as a preconditioned version of S, thanks to block Gaussian elimination. As stated in [4]:

> the overlapping method is equivalent to a non-overlapping method with a specific interface preconditioner. One can think of the overlapping method implicitly computing the effect of this preconditioner by the extra operations performed on the overlapping region.

We observe that in our case, the interface unknowns may not correspond to the interface nodes of the non-overlapping partition. If we consider the permuted matrix $P_\delta M^{-1} A P_\delta^T$, with M being a restricted Schwarz preconditioner, and $P_\delta^T = \left[\bar{R}_\delta^{ext\,T} R_\delta^{ext\,T}\right]$, we get the following linear system:

$$P_\delta M^{-1} A P_\delta^T \begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{Bmatrix} \bar{R}_\delta^{ext} g \\ h \end{Bmatrix} \tag{14}$$

We note that the matrix $P_\delta M^{-1} A P_\delta^T$ is a 2×2 block matrix:

$$P_\delta M^{-1} A P_\delta^T = \begin{bmatrix} \bar{R}_\delta^{ext} M^{-1} A \bar{R}_\delta^{ext\,T} & \bar{R}_\delta^{ext} M^{-1} A R_\delta^{ext\,T} \\ R_\delta^{ext} M^{-1} A \bar{R}_\delta^{ext\,T} & R_\delta^{ext} M^{-1} A R_\delta^{ext\,T} \end{bmatrix} \tag{15}$$

In the previous section, we saw that $F \bar{R}_\delta^{ext\,T} = 0$, which implies that

$$\bar{R}_\delta^{ext} M^{-1} A \bar{R}_\delta^{ext\,T} = I \tag{16}$$

$$R_\delta^{ext} M^{-1} A \bar{R}_\delta^{ext\,T} = 0 \tag{17}$$

We also have the following equalities:

$$\bar{R}_\delta^{ext} M^{-1} A R_\delta^{ext\,T} = -\bar{R}_\delta^{ext} F R_\delta^{ext\,T} \tag{18}$$

$$R_\delta^{ext} M^{-1} A R_\delta^{ext\,T} = I - R_\delta^{ext} F R_\delta^{ext\,T} = I - G \tag{19}$$

Plugging these equalities into (14), we get:

$$P_\delta M^{-1} A P_\delta^T \begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} I & -\bar{R}_\delta^{ext} F R_\delta^{ext\,T} \\ 0 & I - G \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{Bmatrix} \bar{R}_\delta^{ext} g \\ h \end{Bmatrix} \tag{20}$$

The matrix $I - G$ can be seen as a Schur complement of $P_\delta M^{-1} A P_\delta^T$ with respect to the identity operator applied to the non-interface nodes. The inverse of $P_\delta M^{-1} A P_\delta^T$ can be expressed in this way:

$$\left(P_\delta M^{-1} A P_\delta^T\right)^{-1} = \begin{bmatrix} I & \bar{R}_\delta^{ext} F R_\delta^{ext\,T} (I-G)^{-1} \\ 0 & (I-G)^{-1} \end{bmatrix} \tag{21}$$

Also, equation (20) gives us some information about the spectrum of $I - G$:

$$\sigma\left(M^{-1} A\right) = \sigma\left(P_\delta M^{-1} A P_\delta^T\right) = \sigma(I) \cup \sigma(I - G) \tag{22}$$

The spectrum of $I-G$ is the spectrum of $M^{-1}A$ augmented with the eigenvalue 1, which has a multiplicity of $n-n_\delta^{ext}$.

We remark that the cost of explicitly building the $I-G$ matrix is prohibitive, regarding the significant resources required. In the RAS case, the matrix G writes:

$$G = -R_\delta^{ext} \sum_{i=1}^{p} \tilde{R}_{i,\delta}^T A_{i,\delta}^{-1} A_{i,\delta}^{ext} R_{i,\delta}^{ext} R_\delta^{ext\ T} \tag{23}$$

This represents $|\mathcal{V}_{i,\delta}^{ext}|$ local systems to solve for each sub-domain, which solution is dense. This is why iterative methods are preferred.

4 Krylov acceleration

Since matrix A is assumed to be unsymmetrical and indefinite, the GMRES Krylov sub-space method [8] is used to accelerate the iteration (6), as proposed in [1]. The GMRES method is chosen over some other Krylov techniques for its monotonous convergence property. The algorithm used to solve the interface system is presented next, in a left-preconditioned version.

Algorithm 1 GMRES resolution of $(I-G)y = h$

$r_0 = R_\delta^{ext} M^{-1}(b - Ax_0)$, $\beta = \|r_0\|$, and $v_1 = r_0/\beta$
for $j = 1, \ldots, m$ **do**
  $w \leftarrow R_\delta^{ext} M^{-1} A R_\delta^{ext\ T} v_j$
  for $i = 1, \ldots, j$ **do**
    $h_{i,j} \leftarrow (w, v_i)$
    $w \leftarrow w - h_{i,j} v_i$
  end for
  ...
end for
...
Compute $z_m = \text{argmin}_z \|\beta e_1 - \bar{H}_m z\|$ and $y_m = R_\delta^{ext} x_0 + V_m z_m$
If satisfied $y^{(\infty)} \leftarrow y_m$ else restart with $x_0 = R_\delta^{ext\ T} y_m$

An important point is that **Algorithm 1** only differs from the usual one by the use of the restriction and prolongation operators R_δ^{ext} and $R_\delta^{ext\ T}$. Also, one extra step is required to solve the global solution from the interface solution $y^{(\infty)}$:

$$u^{(\infty)} = (I - M^{-1}A) R_\delta^{ext\ T} y^{(\infty)} + g \tag{24}$$

In this last step, the preconditioner M^{-1} can differ from the one used in the GMRES algorithm. For example, if $M_{RAS,\delta}^{-1}$ is chosen, we get:

$$u^{(\infty)} = \sum_{i=1}^{p} \tilde{R}_{i,\delta}^{T} A_{i,\delta}^{-1} R_{i,\delta} \left(b - A \bar{R}_{i,\delta}^{T} \bar{R}_{i,\delta} R_{\delta}^{ext\ T} y^{(\infty)} \right)$$
$$= \sum_{i=1}^{p} \tilde{R}_{i,\delta}^{T} A_{i,\delta}^{-1} \left(R_{i,\delta} b - A_{i,\delta}^{ext} R_{i,\delta}^{ext} R_{\delta}^{ext\ T} y^{(\infty)} \right) \quad (25)$$

Algorithm 1 represents less floating point operations and also requires less memory to store the Arnoldi vectors than when GMRES is applied to the primary unknowns, with almost no extra work regarding the implementation.

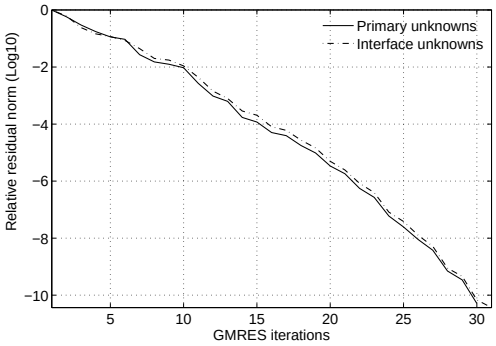


Fig. 1 Full GMRES convergence of the global and interface systems. GT01R matrix from the UF sparse matrix collection is used. Initial condition is $x^{(0)} = \{1, ..., 1\}^T$. The domain is divided into 2 parts ($p = 2$) with an overlap of $\delta = 1$ (all the adjacent nodes are included). The number of primary and interface unknowns is 7980 and 420 respectively.

Fig. 1 presents the GMRES convergence of both primary and interface systems. Matrix GT01R from the UF sparse matrix collection [5] is used. We observe that the convergence behaviors are similar, but slightly differ because of the non-interface nodes. The size of the global system is 7980, while it is 420 for the interface system.

Also, the new vector $w \leftarrow R_{\delta}^{ext} M^{-1} A R_{\delta}^{ext\ T} v_j$ in the outer loop of **Algorithm 1** is equivalent to this one: $w \leftarrow (I - R_{\delta}^{ext} F R_{\delta}^{ext\ T}) v_j$, in which only local "homogeneous" problems are solved. For example in the RAS case, we have:

$$w \leftarrow \left(I + R_{\delta}^{ext} \sum_{i=1}^{p} \tilde{R}_{i,\delta}^{T} A_{i,\delta}^{-1} A_{i,\delta}^{ext} R_{i,\delta}^{ext} R_{\delta}^{ext\ T} \right) v_j \quad (26)$$

The local operator $A_{i,\delta}^{-1}$ is applied to $A_{i,\delta}^{ext} R_{i,\delta}^{ext} R_{\delta}^{ext\ T} v_j$, which only concerns the local interface nodes of the sub-domain, $\mathcal{V}_{i,\delta}^{loc}$. This means that for the local problem in (26), the right-hand side is null for the internal nodes $\mathcal{V}_{i,\delta}^{int}$. Thus, a local Schur complement approach may be used to deal with each local problem, associated to an iterative local solver and the LU factorization of the two diagonal blocks of $A_{i,\delta}$ corresponding to the internal and the local interface nodes.

5 Conclusion

The restricted Schwarz iterations have been described in details. It appears that the restricted Schwarz operators benefit from the indirect preconditioning effect of the overlap, but also from the non-overlapping property of the restricted local operator images. We have seen that solving the interface system instead of the primary one, is advantageous regarding memory usage and floating point operation count. This represents only a slight modification of the global algorithm, but requires exact local solves. Another advantage is that the local problems can be treated as homogeneous problems.

References

1. Brakkee, E., Wilders, P.: A domain decomposition method for the advection-diffusion equation. Tech. rep., Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft (1994)
2. Cai, X.C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J. Sci. Comput. **21**(2), 792–797 (electronic) (1999)
3. Chan, T.F., Goovaerts, D.: On the relationship between overlapping and nonoverlapping domain decomposition methods. SIAM J. Matrix Anal. Appl. **13**(2), 663–670 (1992)
4. Chan, T.F., Mathew, T.P.: Domain decomposition algorithms. In: Acta numerica, 1994, Acta Numer., pp. 61–143. Cambridge Univ. Press, Cambridge (1994)
5. Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Software **38**(1), Art. 1, 25 (2011)
6. Frommer, A., Nabben, R., Szyld, D.B.: An algebraic convergence theory for restricted additive and multiplicative Schwarz methods. In: N. Debit, M. Garbey, R. Hoppe, D. Keyes, Y. Kuznetzov, J. Périaux (eds.) Domain Decomposition Methods in Science and Engineering, Thirteenth International Conference on Domain Decomposition Methods, Lyon, France, pp. 371–377. CIMNE, UPC, Barcelona (2002)
7. Frommer, A., Szyld, D.B.: An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. SIAM J. Numer. Anal. **39**(2), 463–479 (electronic) (2001)
8. Saad, Y.: Iterative methods for sparse linear systems, second edn. Society for Industrial and Applied Mathematics, Philadelphia, PA (2003)
9. Saad, Y., Sosonkina, M.: Distributed Schur complement techniques for general sparse linear systems. SIAM J. Sci. Comput. **21**(4), 1337–1356 (electronic) (1999/00)
10. Wilders, P., Brakkee, E.: Schwarz and Schur: an algebraical note on equivalence properties. SIAM J. Sci. Comput. **20**(6), 2297–2303 (electronic) (1999)

A nonlinear domain decomposition technique for scalar elliptic PDEs

James Turner[1], Michal Kočvara[1], and Daniel Loghin[1]

1 Introduction

Nonlinear problems are ubiquitous in a variety of areas, including fluid dynamics, biomechanics, viscoelasticity and finance, to name a few. A number of computational methods exist already for solving such problems, with the general approach being Newton-Krylov type methods coupled with an appropriate preconditioner. However, it is known that the strongest nonlinearity in a domain can directly impact the convergence of Newton-type algorithms. Therefore, local nonlinearities may have a direct impact on the global convergence of Newton's method, as illustrated in both [3] and [5]. Consequently, Newton-Krylov approaches can be expected to struggle when faced with domains containing local nonlinearities.

An attempt to resolve this issue was considered in [4] by Cai and Li. Here, a method based on an overlapping decomposition of the domain was proposed, which involved the development of a nonlinear restrictive additive Schwarz preconditioner for the treatment of high nonlinearities. Effectively, their proposed method ensured that the distribution of nonlinearities was balanced throughout their system, building on earlier work in [9]. While positive results were obtained, it is noted that their numerical experiments display a logarithmic dependence with regard to the mesh size. Additionally, in the situation of the unavailability of sufficient processors, it was found that subdomain problems could become computationally demanding, due in part to the need for a region of overlap. An alternative approach would be to instead consider applying a nonoverlapping decomposition of the domain directly to the nonlinear problem, avoiding the linearisation on a global scale. Methods have been proposed to this effect by both Pebrel et. al. [12] and by Sassi [14]. In [12], the resulting algorithm involved the solution to local nonlinear subproblems, as well as a global interface problem solved by a Newton-type algorithm. As a result, local nonlinearities could be dealt with much more effectively without having a major impact on the solution across the whole domain. While the paper reported speed up in the CPU time when compared directly to a Newton-Krylov approach, the method proposed involves the solution of a global interface problem, which can be both expensive and time consuming to compute. In comparison, [14] considered a preconditioned modified Newton algorithm, which was found to converge independently of the mesh size. However, the diameter of each subdomain was found to have a direct influence on the condition number of the involved operator, and as a result the proposed algorithm struggled with an increasing number of subdomains.

[1] University of Birmingham, B15 2TT, UK. e-mail: {jat649}{m.kocvara}{d.loghin}@bham.ac.uk

We propose a splitting of a class of nonlinear problems into a three step procedure wrapped around a fixed point iteration. Section 2 will provide a description of the model problem, before the application of domain decomposition to the nonlinear problem in Section 3. A three step procedure can then be devised by applying an appropriate Picard linearisation (Section 4), which will be wrapped inside a global fixed point iteration. The corresponding weak formulation and finite element discretisation of the problem are given in Section 5, with results from the proposed method illustrated in Section 6.

2 Model Problem

We begin by considering the following problem posed on a two dimensional open and simply connected domain Ω:

$$\begin{cases} \mathscr{N}(u) := -\Delta u + c(u) = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where the function $c(u)$ is nonlinear and $\mathscr{N}$ is assumed to be positive. We also assume that (1) has a unique solution. A number of real life situations can be simulated by the nonlinear diffusion equation (1); in particular, notable applications can be found when modelling flow through porous material, in biochemistry, and in the transport of radiation.

An established approach for dealing with problems of type (1) is to employ Newton-Krylov methods and use domain decomposition methods as preconditioners. A number of preconditioning strategies have been considered (e.g. additive-Schwarz [7, 11], approximate-Schur [8, 13]), giving rise to numerous different Newton-Krylov type approaches, which have been applied to a wide range of problems mainly due to the quadratic convergence of Newton's method. However, for domains containing high local nonlinearities, the global convergence of Newton's method becomes entirely dependent on the local phenomena contained within the domain. Therefore, a substantial number of iterations can be expected for certain problems solved using such approaches, even for domains containing predominantly smooth areas, and so it is desirable to consider alternative approaches for determining solutions to systems of the form (1).

3 Nonlinear Domain Decomposition

We consider an approach that applies domain decomposition directly to the nonlinear problem. To do this, we divide our domain Ω into N nonoverlapping subdomains Ω_i with boundary $\partial\Omega_i$ with outer normals $\mathbf{n}_i$. We denote by Γ the resulting skeletal interface $\Gamma = \bigcup_{i=1}^{N} \Gamma_i$, where $\Gamma_i := \partial\Omega_i \backslash \partial\Omega$. The restriction of a function w to a subdomain Ω_i is denoted by w_i. Assuming $u_i|_{\Gamma_i} = \lambda_i$ is given, problem (1) can then be seen to be equivalent to the following subproblems

$$\begin{cases} \mathscr{N}(u_i) := -\Delta u_i + c(u_i) = f_i & \text{in } \Omega_i \\ u_i = 0 & \text{on } \partial\Omega_i \backslash \Gamma_i \\ u_i = \lambda_i & \text{on } \Gamma_i. \end{cases}$$

Let $u = u^1 + u^2$ and assume that the nonlinear function $c(u)$ can be written as $c(u^1+u^2) = c^1(u^1+u^2) + c^2(u^1+u^2)$. The reason for splitting u and c in this way is to attempt to form homogeneous Dirichlet subdomain problems around u^1. The remaining components will then form subdomain problems around u^2.

Problem (1) can be viewed in terms of the following subproblems

$$\begin{cases} -\Delta u_i^1 + c^1(u_i^1 + u_i^2) = f_i & \text{in } \Omega_i \\ u_i^1 = 0 & \text{on } \partial\Omega_i \end{cases} \tag{2a}$$

$$\begin{cases} \sum_{i=1}^{N} \left(\mathbf{n}_i \cdot \nabla(u_i^2)\right) = -\sum_{i=1}^{N} \left(\mathbf{n}_i \cdot \nabla(u_i^1)\right) & \text{on } \Gamma \end{cases} \tag{2b}$$

$$\begin{cases} -\Delta u_i^2 + c^2(u_i^1 + u_i^2) = 0 & \text{in } \Omega_i \\ u_i^2 = 0 & \text{on } \partial\Omega_i \backslash \Gamma_i \\ u_i^2 = \lambda_i & \text{on } \Gamma_i. \end{cases} \tag{2c}$$

The nonlinear subproblems presented in (2a) correspond to obtaining solutions to local copies of (1) with homogeneous Dirichlet conditions enforced on local boundaries $\partial\Omega_i$. In comparison, the nonlinear subdomain problems presented in (2c) use interfacial data found in the intermediate step (2b) to obtain local solutions. The main motivation for considering such a splitting, and indeed for considering a nonoverlapping decomposition of Ω is that each subproblem in both (2a) and (2c) can be solved independently of other subdomains. In the following, we will assume that solution operators exist for problems of the form (2c); these will be denoted by E_i; in particular, we have $u_i^2 = E_i(\lambda_i)$. We will denote by $F_i\mu_i$ any other linear extensions of a given function μ_i defined on Γ_i to Ω_i.

4 Picard Linearisation

We decouple (2a), (2b) and (2c) via the following Picard linearisation

$$\begin{cases} \mathcal{N}_1(u_i^{1,k}) := -\Delta u_i^{1,k} + c^1\left(u_i^{1,k} + u_i^{2,k-1}\right) = f_i & \text{in } \Omega_i \\ u_i^{1,k} = 0 & \text{on } \partial\Omega_i \end{cases} \tag{3a}$$

$$\begin{cases} \sum_{i=1}^{N} \mathbf{n}_i \cdot \nabla(E_i^{k-1}\lambda_i^k) = -\sum_{i=1}^{N} \mathbf{n}_i \cdot \nabla(u_i^{1,k}) & \text{on } \Gamma \end{cases} \tag{3b}$$

$$\begin{cases} \mathcal{N}_2(u_i^{2,k}) := -\Delta u_i^{2,k} + c^2\left(u_i^{1,k} + u_i^{2,k}\right) = 0 & \text{in } \Omega_i \\ u_i^{2,k} = 0 & \text{on } \partial\Omega_i \backslash \Gamma_i \\ u_i^{2,k} = \lambda_i^k & \text{on } \Gamma_i. \end{cases} \tag{3c}$$

Given u^{k-1}, N nonlinear subproblems are first solved independently in (3a). The solution to these subproblems is then used in equation (3b) to obtain the interface update λ_i^k. Finally, the solutions to each nonlinear subproblem in (3c) are obtained independently using the updates from the previous two steps. Note that it is pos-

sible to solve each of the two sets of N nonlinear subproblems in (3a) and (3c) in parallel. Equation (3b) is a linear Steklov-Poincaré equation involving the operator $S^k\colon H_{00}^{1/2}(\Gamma) \to H_{00}^{-1/2}(\Gamma)$ defined as

$$\langle S^k\lambda^k, \mu\rangle := \sum_{i=1}^{N} \int_{\Gamma_i} (\mathbf{n}_i \cdot \nabla)(E_i^{k-1}\lambda_i)\mu_i \,\mathrm{ds} = \sum_{i=1}^{N} \langle S_i^k \lambda_i^k, \mu_i\rangle,$$

where E_i^{k-1} are linearizations of the nonlinear extension operators E_i corresponding to (3c). We summarize below the proposed iterative scheme for computing the exact solution u^*, given an initial u^0.

(i) Run through the following three steps to compute the solution $u^k = u^{1,k} + u^{2,k}$:

$$\begin{cases} \mathscr{N}_1(u_i^{1,k}) = f & \text{in } \Omega_i \\ u_i^{1,k} = 0 & \text{on } \partial\Omega_i \end{cases} \qquad i = 1,\dots,N. \tag{4a}$$

$$\begin{cases} S^k\lambda^k = -\sum_{i=1}^{N} \mathbf{n}_i \cdot \nabla(u_i^{1,k}) & \text{on } \Gamma \end{cases} \tag{4b}$$

$$\begin{cases} \mathscr{N}_2(u_i^{2,k}) = 0 & \text{in } \Omega_i \\ u_i^{2,k} = 0 & \text{on } \partial\Omega_i \backslash \Gamma_i \\ u_i^{2,k} = \lambda_i^k & \text{on } \Gamma_i. \end{cases} \qquad i = 1,\dots,N. \tag{4c}$$

(ii) Compute the residual $\mathscr{R}^k = \mathscr{N}(u^k) - f$. If $\|\mathscr{R}^k\| < \tau$, set $u^* = u^k$ and terminate. Else, set $k = k+1$ and return to step 1.

5 Finite Element Discretisation

Define now local bilinear forms

$$a_i^l(v,w;z) := \int_{\Omega_i} \nabla v \nabla w \,\mathrm{d}x + \int_{\Omega_i} c^l(v+z)\, w \,\mathrm{d}x,$$

for $l = 1,2$. Using the above notation, the weak formulation of (4) is

$$\begin{cases} \text{Find } u_i^{1,k} \in H_0^1(\Omega_i) \text{ such that } \forall v_i \in H_0^1(\Omega_i) \\ a_i^1(u_i^{1,k}, v_i; u_i^{2,k-1}) = (f_i, v_i) \end{cases} \tag{5a}$$

$$\begin{cases} \text{Find } \lambda^k \in H_{00}^{1/2}(\Gamma) \text{ such that } \forall \mu \in H_{00}^{1/2}(\Gamma) \\ s(\lambda^k, \mu) = \sum_{i=1}^{N} (f_i, F_i\mu_i) - a_i^1(u_i^{1,k}, F_i\mu_i; u_i^{2,k-1}) \end{cases} \tag{5b}$$

$$\begin{cases} \text{Find } u_i^{2,k} \in E(\lambda_i^k) + H_0^1(\Omega_i) \text{ such that } \forall v_i \in H_0^1(\Omega_i) \\ a_i^2(u_i^{2,k}, v_i; u_i^{1,k}) = 0. \end{cases} \tag{5c}$$

Let now $V_h \subset H_0^1(\Omega) \cap C^0(\Omega)$ be a space of continuous piecewise polynomials of degree m defined on an isotropic subdivision of Ω into simplices of maximum diameter h. In our tests we choose $m = 1$, though other values are equally possible. Let the corresponding basis be denoted by $\{\psi_r\}$. Let B denote the index set corresponding to basis elements ψ_r with support on Γ. Let $S_h := \text{span}\{\gamma_0(\Gamma)\,\psi_r : r \in B\}$ where γ_0 denotes the trace operator. The finite element discretisation of the systems in (5) can then be written for $i = 1, \ldots, N$ as

$$\begin{cases} \text{Find } u_{i,h}^{1,k} \in V_{i,h} \text{ such that } \forall v_{i,h} \in V_{i,h} \\ a_i^1(u_{i,h}^{1,k}, v_{i,h}; u_{i,h}^{2,k-1}) = (f_i, v_{i,h}) \end{cases} \tag{6a}$$

$$\begin{cases} \text{Find } \lambda_h^k \in S_h \text{ such that } \forall \mu_h \in S_h \\ s(\lambda_h^k, \mu_h) = \sum_{i=1}^{N} (f_i, F_i \mu_i) - a_i^1(u_{i,h}^{1,k}, F_i \mu_{i,h}; u_{i,h}^{2,k-1}) \end{cases} \tag{6b}$$

$$\begin{cases} \text{Find } u_{i,h}^{2,k} \in (E\lambda)_{i,h}^k + V_{i,h} \text{ such that } \forall v_{i,h} \in V_{i,h} \\ a_i^2(u_{i,h}^{2,k}, v_{i,h}; u_{i,h}^{1,k}) = 0. \end{cases} \tag{6c}$$

The system (6) can be represented systematically by matrices and vectors in the usual way. In particular, the Schur complement of the system matrix corresponds to the matrix representation of $s(\cdot,\cdot)$ in the basis of S_h. We can therefore describe our proposed method as follows:

(i) Run through the following three step procedure to determine $\mathbf{u}$.

a. Solve the N decoupled nonlinear subdomain problems (6a) written in matrix form as

$$A_{II}^{i,1}(\mathbf{u}_{I,i}^{1,k})\mathbf{u}_{I,i}^{1,k} = \mathbf{f}_{I,i}^1, \tag{7a}$$

using a Newton-Krylov method with line search and adaptive tolerances $\tau_{1,i}$.

b. Calculate interface values $\boldsymbol{\lambda}^k$ using

$$S^k \boldsymbol{\lambda}^k = \mathbf{f}_\Gamma - \sum_{i=1}^{N} A_{\Gamma I}^{i,1}(\mathbf{u}_{I,i}^{1,k})\mathbf{u}_{I,i}^{1,k}. \tag{7b}$$

c. Solve the N decoupled nonlinear subdomain problems (6c) written in matrix form as

$$A_{II}^{i,2}(\mathbf{u}_{I,i}^{2,k})\mathbf{u}_{I,i}^{2,k} = -A_{I\Gamma}^{i,2}(\mathbf{u}_{I,i}^{2,k})\boldsymbol{\lambda}_i^k, \tag{7c}$$

using a Newton-Krylov method with line search and adaptive tolerances $\tau_{2,i}$.

(ii) Set $\mathbf{u}^k = \mathbf{u}^{1,k} + \mathbf{u}^{2,k}$, where $\mathbf{u}^{1,k} = [\mathbf{u}_I^{1,k}, 0]^T$ and $\mathbf{u}^{2,k} = [\mathbf{u}_I^{2,k}, \boldsymbol{\lambda}^k]^T$. Assemble the global stiffness matrix $A^k(\mathbf{u})$ and compute the residual $\mathscr{R}^k(\mathbf{u}^k) = A(\mathbf{u}^k)\mathbf{u}^k - \mathbf{f}$. If $\|\mathscr{R}^k\| < \tau$ set $\mathbf{u}^* = \mathbf{u}^k$ and exit; else, return to Step 1.

The subindices I and Γ indicate permutations involving the index sets corresponding to the interior and boundary nodes in the subdivision of Ω. The adaptive tolerances $\tau_{1,i}, \tau_{2,i}$ are chosen in relation to the norm of the global nonlinear residual $\|\mathscr{R}^k\|$, following the strategy in [6].

We solve the system (7b) using iterative methods of Krylov type with preconditioning. The matrix S^k is the interface Schur complement corresponding to the reaction-diffusion problem $-\Delta u^{2,k-1} + c^2(u^{1,k-1} + u^{2,k-1})$; as such, it can be preconditioned by any domain decomposition preconditioner designed for elliptic problems. The preconditioner employed in this work is based on [2], where discrete norms corresponding to finite element discretisations of fractional Sobolev spaces are presented. In particular, it was shown that a discrete norm on $S_h \subset H_{00}^{1/2}(\Gamma)$ which is spectrally equivalent to S^k is given by

$$H_{1/2} = M_\Gamma \left(M_\Gamma^{-1} L_\Gamma\right)^{1/2}.$$

In [2], M_Γ and L_Γ correspond to the mass and Laplacian matrices, respectively, assembled on Γ. We adapt the definition of $H_{1/2}$ to include the contribution from the reaction term as suggested in [1]; this involves replacing L_Γ with

$$L_\Gamma^k = L_\Gamma + M_\Gamma^k,$$

where M_Γ^k is the mass matrix assembled on Γ and weighted by the trace on the interface Γ of $c^2\left(u^{1,k-1} + u^{2,k-1}\right)$. For more details, see [15].

Note that M_Γ, L_Γ^k are assembled globally on Γ and hence $H_{1/2}$ is a dense matrix. However, in our computations we use sparse techniques to circumvent this issue. In particular, the application of both Lanczos and inverse Lanczos factorisations has been considered in [2], and will be applied in this work in a similar manner.

6 Results

In this section, we will consider a number of examples to highlight the benefits of our proposed method. In particular, we will consider models for which

$$\text{(a) } c(u) = u^{q+1}, \qquad \text{and} \qquad \text{(b) } c(u) = u^{q+1}\sin(10u),$$

where q is a positive integer. For both choices, we note that by substituting $u = u^1 + u^2$ into the function, we can write

$$c(u^1+u^2) = (u^1+u^2)^{q+1} = (u^1+u^2)^q u^1 + (u^1+u^2)^q u^2.$$

Table 1 displays performance comparisons of our proposed method to the standard Newton-Krylov approach for two test problems. We used piecewise linear discretizations for a range of mesh parameters h. Each nonlinear problem was solved with a zero initial guess. We consider four different representations for the preconditioner $\tilde{S}$, namely the exact Schur complement, the exact discrete fractional Sobolev norm $H_{1/2}$, and both the Lanczos (L) and inverse Lanczos (I) approximations to $H_{1/2}$. The performance recorded in the table indicates that our method delivers promising results when directly compared to the corresponding Newton-Krylov

		(a), $q=2$						(b), $q=9$					
		Newton-Krylov			3-Step Procedure			Newton-Krylov			3-Step Procedure		
$\tilde{S}$	h	4	16	64	4	16	64	4	16	64	4	16	64
S	1/16	4 (8)	4 (8)	3 (6)	4 (8)	4 (8)	3 (5)	11 (22)	10 (20)	10 (20)	6 (12)	5 (10)	5 (10)
	1/32	4 (8)	3 (6)	3 (6)	4 (8)	3 (6)	3 (5)	10 (20)	10 (20)	9 (18)	5 (10)	5 (10)	5 (10)
	1/64	3 (6)	3 (6)	3 (6)	3 (6)	3 (6)	3 (5)	10 (20)	9 (18)	8 (16)	5 (10)	5 (10)	4 (8)
$H_{1/2}$	1/16	9 (24)	11 (35)	6 (48)	6 (29)	6 (26)	5 (35)	15 (66)	16 (89)	12 (110)	7 (48)	8 (60)	6 (47)
	1/32	12 (38)	7 (42)	5 (49)	6 (36)	5 (25)	4 (30)	17 (79)	12 (103)	10 (110)	7 (53)	6 (49)	5 (52)
	1/64	6 (33)	4 (29)	4 (42)	5 (28)	4 (25)	4 (41)	11 (75)	9 (96)	8 (103)	6 (47)	4 (28)	4 (43)
$H^{(L)}_{1/2}$	1/16	9 (24)	11 (35)	6 (48)	6 (29)	6 (26)	5 (35)	15 (66)	16 (89)	12 (110)	7 (48)	8 (60)	6 (47)
	1/32	12 (38)	7 (42)	5 (49)	6 (36)	5 (25)	4 (30)	17 (79)	12 (103)	10 (110)	7 (53)	6 (49)	5 (52)
	1/64	6 (33)	4 (29)	4 (42)	5 (28)	4 (25)	4 (41)	11 (75)	9 (96)	8 (103)	6 (47)	4 (28)	4 (43)
$H^{(I)}_{1/2}$	1/16	17 (39)	106 (148)	7 (51)	6 (31)	6 (34)	5 (39)	21 (78)	132 (193)	12 (102)	8 (64)	7 (56)	6 (58)
	1/32	11 (40)	6 (35)	4 (36)	6 (37)	5 (35)	4 (33)	14 (77)	11 (92)	5 (53)	7 (54)	6 (49)	5 (55)
	1/64	6 (37)	4 (34)	4 (40)	5 (36)	4 (27)	3 (27)	11 (76)	9 (90)	8 (92)	5 (40)	5 (44)	4 (45)

Table 1 Nonlinear iterations (total GMRES iterations) for a global tolerance $\tau = 10^{-7}$.

method. In particular, it can be seen that the results indicate independence with respect to both the mesh size and the number of subdomains used. By comparing the columns in Table 1, an indication is given on how well both methods adapt to the increase in nonlinearity. Notably, it is clear that the Newton-Krylov method struggled when faced with the increased nonlinearity, confirming results noted earlier. However, in comparison our method was found to deal with the increase in nonlinearity in a much more efficient manner. This would suggest that our method would adapt quite well to domains containing high local nonlinearities confined to a particular region of the domain. It is also noted that by directly inverting the Schur complement, an adaptation of the result presented in [10] is also shown for our method, namely that the interface problem (7b) solved with GMRES can be expected to converge in a number of iterations no more than the dimension of Ω per fixed point iteration.

7 Conclusion

In this paper, we introduced a three step procedure for solving a class of nonlinear PDEs. We have demonstrated that our method is able to deliver results independent of both the mesh size and the number of subdomains used. Furthermore, we have shown that our procedure is competitive when directly compared to the corresponding Newton-Krylov method. Future work will involve further testing to include problems that contain a high nonlinearity confined to a particular region of the domain together with an appropriate analysis of the method. We will also adapt our method to problems in topology optimization [15].

Acknowledgements Michal Kočvara would like to acknowledge the financial support provided by the Grant Agency of the Czech Republic through project GAP201-12-0671. Additionally, James Turner would like to acknowledge the financial support provided by the University of Birmingham.

References

1. Arioli, M., Kourounis, D., Loghin, D.: Discrete fractional Sobolev norms for domain decomposition preconditioning. IMA J. Numer. Anal. (2012). doi:10.1093/imanum/drr024
2. Arioli, M., Loghin, D.: Discrete interpolation norms with applications. SIAM J. Numer. Anal. **47**(4), 2924–2951 (2009)
3. Cai, X.C., Keyes, D.E.: Nonlinearly preconditioned inexact Newton algorithms. SIAM Journal on Scientific Computing **24**(1), 183–200 (2002)
4. Cai, X.C., Li, X.: Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity. SIAM Journal on Scientific Computing **33**(2), 746–762 (2011)
5. Cresta, P., Allix, O., Rey, C., Guinard, S.: Nonlinear localization strategies for domain decomposition methods: application to post-buckling analyses. Computer Methods in Applied Mechanics and Engineering **196**(8), 1436–1446 (2007)
6. Dembo, R.S., Eisenstat, S.C., Steihaug, T.: Inexact Newton methods. SIAM Journal on Numerical Analysis **19**(2), 400–408 (1982)
7. Groth, C.P.T., Northrup, S.A.: Parallel implicit adaptive mesh refinement scheme for body-fitted multi-block mesh. In: 17th AIAA Computational Fluid Dynamics Conference, Toronto, Ontario, Canada, AIAA paper, vol. 5333, p. 2005 (2005)
8. Hicken, J.E., Zingg, D.W.: Parallel Newton-Krylov solver for the Euler equations discretized using simultaneous-approximation terms. AIAA journal **46**(11), 2773 (2008)
9. Hwang, F.N.A.N., Lin, H.L.U.N., Cai, X.C.: Two-level nonlinear elimination based preconditioners for intexact Newton mathods with application in shocked duct flow calculation. Electronic Transactions on Numerical Analysis **37**, 239–251 (2010)
10. Ipsen, I.: A note on preconditioning nonsymmetric matrices. SIAM Journal of Scientific Computing **23**(3), 1050–1051 (2002)
11. Kaushik, D.K., Keyes, D.E., Smith, B.F., et al.: Newton-Krylov-Schwarz methods for aerodynamic problems: Compressible and incompressible flows on unstructured grids. In: Proceedings of the 11th International Conference on Domain Decomposition Methods. Domain Decomposition Press, Bergen (1999)
12. Pebrel, J., Rey, C., Gosselet, P.: A nonlinear dual domain decomposition method: application to structural problems with damage. International Journal of Multiscale Computational Engineering **6**(3), 251–262 (2008)
13. Saad, Y., Sosonkina, M.: Distributed Schur complement techniques for general sparse linear systems. SIAM Journal on Scientific Computing **21**(4), 1337–1356 (1999)
14. Sassi, T.: A domain decomposition algorithm for nonlinear interface problem. In: Domain decomposition methods in science and engineering, pp. 467–474 (electronic). Natl. Auton. Univ. Mex., México (2003)
15. Turner, J.A.: Application of Domain Decomposition to problems in Topology Optimization. Ph.D. thesis, University of Birmingham (Submission date: 2013/14)

A non overlapping domain decomposition method for the obstacle problem

Samia Riaz[1] and Daniel Loghin[2]

1 Obstacle problem

The obstacle problem is to determine the equilibrium position of an elastic membrane in a domain $\Omega \subseteq \mathbb{R}^2$ with closed boundary $\partial\Omega$, which lies above an obstacle function $\psi : \Omega \to \mathbb{R}^+$ under the vertical force f. The classical solution u of this model problem is the vertical displacement of the membrane. Since the membrane is fixed on $\partial\Omega$, we have boundary conditions of Dirichlet type (say $u = 0$). The problem can be written as

$$\begin{cases} -\Delta u - f \geq 0 & \text{in } \Omega, \\ u - \psi \geq 0 & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{1}$$

subject to the pointwise complementarity condition $(u-\psi)(-\Delta u - f) = 0$.
Let $\mathscr{C} = \{\mathbf{x} \in \Omega : u(\mathbf{x}) = \psi(\mathbf{x})\}$ denote the coincidence set. Then the complementarity conditions yields the PDE $-\Delta u - f = 0$ in $\Omega \setminus \mathscr{C}$. The weak formulation of (1) can be written as [10]

$$\begin{cases} \text{Find } u \in K \text{ such that } \forall v \in K, \\ a(u, v-u) \geq (f, v-u), \end{cases} \tag{2}$$

which can be shown to be equivalent to the following minimization problem

$$\begin{cases} \text{Find } u \in K, \text{ such that } \forall\, v \in K, \\ J(u) \leq J(v), \end{cases}$$

where $K = \{v \in V := H_0^1(\Omega) : v \geq \psi \text{ in } \Omega\}$ is convex and

$$J(v) = \frac{1}{2}a(\nabla v, \nabla v) - (f, v), \quad (f, v) = \int_\Omega f v d\Omega, \quad a(u, v) = \int_\Omega \nabla u \cdot \nabla v d\Omega.$$

An important class of solution techniques for (2) is that of multilevel and multigrid methods for constrained minimization problems, first introduced by [6] and [2] some variants of these method were studied in [4] and were analyzed in [5]. A challenging task for multigrid is the representation of the coincidence set on a coarse grid, as shown in the review paper [3]. Some multi-grid and two level domain decomposition methods are given in [9] [7] in which it is shown that the overlapping DDM has a

[1] University of Birmingham, UK and UET, Lahore, Pakistan e-mail: riazs@for.mat.bham.ac.uk .[2] University of Birmingham, UK daniel.loghin@bham.ac.uk

linear convergence for constrained obstacle problem if the obstacle and computed functions decomposed properly. Some more variants of multi-grid method are given in [1], where the decomposition of the closed convex set for minimization problem is introduced as a sum of closed convex level subsets; the convergence rate is shown to depend on the number of levels.

2 A non-overlapping domain decomposition method

Let Ω^i denote an open subset of Ω containing the coincidence set $\mathscr{C}$ and let $\Omega^e = \Omega \setminus \Omega^i$. Let Γ denote the interface between Ω^i and Ω^e. This decomposition allows us to reformulate our problem into two subproblems: one which is a partial differential inequality (PDI) in subdomain Ω^i and the other which is a partial differential equation (PDE) in Ω^e.
Let $z = u|_{\Omega^e}, w = u|_{\Omega^i}$, $f^e = f|_{\Omega^e}$ and $f^i = f|_{\Omega^i}$ be the restrictions of u and f to Ω^e and Ω^i respectively; let also $\lambda = u|_\Gamma$ be the trace of u on Γ. Assuming for now that λ is known, problem (1) decouples into the two subproblems

$$\text{PDE:} \begin{cases} -\Delta z = f^e & \text{in } \Omega^e, \\ z = 0 & \text{on } \partial\Omega \setminus \Gamma, \\ z = \lambda & \text{on } \Gamma, \end{cases} \qquad \text{PDI:} \begin{cases} -\Delta w \geq f^i & \text{in } \Omega^i, \\ w \geq \psi^i & \text{in } \Omega^i, \\ w = 0 & \text{on } \partial\Omega \setminus \Gamma, \\ w = \lambda & \text{on } \Gamma. \end{cases}$$

with $(-\Delta w - f^i)(w - \psi^i) = 0$ satisfied in a pointwise sense in Ω^i. The subproblem PDE can be further decoupled as follows:

$$PDE_1 : \begin{cases} -\Delta z_1 = f^e & \text{in } \Omega^e, \\ z_1 = 0 & \text{on } \partial\Omega \setminus \Gamma, \\ z_1 = 0 & \text{on } \Gamma, \end{cases} \quad PDE_2 : \begin{cases} -\Delta z_2 = 0 & \text{in } \Omega^e, \\ z_2 = 0 & \text{on } \partial\Omega \setminus \Gamma, \\ z_2 = \lambda & \text{on } \Gamma, \end{cases} \tag{3}$$

where $z|_{\Omega^e} = z_1 + z_2$ with $z_2 = E\lambda$ where E is the harmonic extension operator to Ω^e. Writing the weak formulation (2) as

$$a^e(z, v-z) + a^i(w, v-w) \geq (f^e, v-z)_{\Omega^e} + (f^i, v-w)_{\Omega^i}, \tag{4}$$

where

$$a^e(z,v) = \int_{\Omega^e} \nabla z \cdot \nabla v \, d\Omega^e \text{ and } a^i(w,v) = \int_{\Omega^i} \nabla w \cdot \nabla v \, d\Omega^i$$

the variational formulations of (3) and PDI are

$$\begin{cases} \text{find } z_1 \in H_0^1(\Omega^e) \text{ such that } \forall v \in H_0^1(\Omega^e) \\ a^e(z_1, v-z) - \displaystyle\int_\Gamma \mathbf{n}_1 \cdot \nabla z_1 \cdot (v-z) \, d\Gamma = (f^e, v-z)_{\Omega^e}, \end{cases}$$

$$\begin{cases} \text{find } z_2 \in H^1(\Omega^e) \text{ such that } \ \forall v \in H^1(\Omega^e) \\ a^e(z_2, v-z) - \displaystyle\int_\Gamma \mathbf{n}_1 \cdot \nabla z_2 \cdot (v-z)\, d\Gamma = 0, \end{cases} \tag{5}$$

$$\begin{cases} \text{find } w \in H^1(\Omega^i) \text{ such that } \ \forall v \in H^1(\Omega^i) \\ a^i(w, v-w) - \displaystyle\int_\Gamma \mathbf{n}_2 \cdot \nabla w \cdot (v-w) d\Gamma \geq (f^i, v-w)_{\Omega^i}. \end{cases} \tag{6}$$

For $i = 1,2$, $\mathbf{n}_i$, is the normal direction from Ω^e and Ω^i respectively. Adding the above weak formulations, where $z_1 = 0$, $z_2 = \lambda = w$ on Γ and using the weak formulation (4) yields a partial Steklov-Poincaré inequality for λ (corresponding to the splitting of PDE)

$$(\mathscr{S}^e \lambda, \mu - \lambda) \leq (g(\lambda), \mu - \lambda).$$

Using the assumption that the interface Γ lies outside the support of the obstacle we obtain the following nonlinear equation on the interface

$$(\mathscr{S}^e \lambda, \mu) = (g(\lambda), \mu). \tag{7}$$

The Steklov-Poincaré operator $\mathscr{S}^e : \Lambda \to \Lambda'$ (where $\Lambda = H^{1/2}(\Gamma)$, $H_0^{1/2}(\Gamma)$ or $H_{00}^{1/2}(\Gamma)$ depending on the nature of the problem) is defined as

$$(\mathscr{S}^e \lambda, \mu) \ := \int_\Gamma (\mathbf{n_1} \cdot \nabla (E\lambda)) \mu \, d\Gamma,$$

and

$$(g(\lambda), \mu) \ := -\int_\Gamma (\mathbf{n_1} \cdot \nabla z_1 + \mathbf{n_2} \cdot \nabla w) \mu \, d\Gamma$$

Applying Green's formula we get the alternative representation of $\mathscr{S}^e$

$$(\mathscr{S}^e \lambda, \mu) := a^e(E\lambda, F\mu) \quad \forall \lambda, \mu \in \Lambda$$

where F denotes an arbitrary extension operator to Ω^e. By using the above definition of $\mathscr{S}^e$, our classical problem can be written as an ordered sequence of three decoupled problems involving Poisson problem on subdomain Ω^e together with a problem set on the interface Γ which is coupled with the problem on Ω^i.

$$\begin{cases} -\Delta z_1 = f^e & \text{in } \Omega^e, \\ z_1 = 0 & \text{on } \partial\Omega \setminus \Gamma, \\ z_1 = 0 & \text{on } \Gamma, \end{cases}$$

$$(i) \left\{ \mathscr{S}^e \lambda = -\mathbf{n_1} \cdot \nabla \mathbf{z_1} - \mathbf{n_2} \cdot \nabla \mathbf{w}, \right. \qquad (ii) \begin{cases} -\Delta w \geq f^i & \text{in } \Omega^i, \\ w \geq \psi & \text{in } \Omega^i, \\ w = 0 & \text{on } \partial\Omega \setminus \Gamma, \\ w = \lambda & \text{on } \Gamma, \end{cases}$$

$$\begin{cases} -\Delta z_2 = 0 & \text{in } \Omega^e, \\ z_2 = 0 & \text{on } \partial\Omega \setminus \Gamma, \\ z_2 = \lambda & \text{on } \Gamma, \end{cases}$$

The resulting solution in Ω^e, is $u|_{\Omega^e} = z = z_1 + z_2$. The solutions of $(i), (ii)$, i.e. λ and w can be approximated in an iterative manner by using a fixed point iteration (see section 2.3). The weak formulations of the above problems are given below.

$$\begin{cases} \text{find } z_1 \in H_0^1(\Omega^e) \text{ such that } \forall v \in H_0^1(\Omega^e), \\ a^e(z_1, v) = (f^e, v)_{\Omega^e}, \end{cases} \tag{8a}$$

$$\begin{cases} \text{find } \lambda \in \Lambda \text{and } w \in E\lambda + K^i \text{ such that } \forall \mu \in \Lambda \text{and } v \in K^i, \\ (\mathscr{S}^e\lambda, \mu) = ((f^e, F^e\mu^e) - a^e(z_1, F^e\mu^e)) + ((f^i, F^i\mu^i) - a^i(w, F^i\mu^i)), \\ a^i(w, v-w) \geq (f^i, v-w)_{\Omega^i}, \end{cases} \tag{8b}$$

$$\begin{cases} \text{find } z_2 \in E\lambda + H_0^1(\Omega^e) \text{ such that } \forall v \in H_0^1(\Omega^e), \\ a^e(z_2, v) = 0, \end{cases} \tag{8c}$$

where $K^i = \{v \in V := H_0^1(\Omega^i) : \ v \ \geq \ \psi\}$.

2.1 Finite element discretization

Let $\Omega \subset \mathbb{R}^2$ be a bounded open convex subset and let $\mathfrak{T}_h$ be a conforming isotropic subdivision of $\overline{\Omega}$ into simplices $\mathbf{t}$. Let V_h^e, V_h^i denote the spaces of continuous piecewise polynomials defined on the corresponding subdivision of Ω^e, Ω^i.

$$K_h^e := \{v_h \in V_h^e : v_h|_{\partial\Omega^e \cap \partial\Omega} = 0\}, \quad K_h^i := \{v_h \in V_h^i : v_h \geq \psi, \ v|_{\partial\Omega^i \cap \partial\Omega} = 0\}.$$

Let $\mathscr{N}^e, \mathscr{N}^i, \mathscr{N}^\Gamma$ denote the sets of nodes located, respectively, in the subdomains Ω^e, Ω^i and on the interface Γ. Let

$$K_h^e = \text{span}\{\phi_k, \ k \in \mathscr{N}^e\}, \ \ K_h^i = \text{span}\{\phi_k, \ k \in \mathscr{N}^i\} \ \ , K_h^\Gamma = \text{span}\{\phi_k, \ k \in \mathscr{N}^\Gamma\}$$

and let

$$S^h = \text{span}\{\gamma_0(\Gamma)\phi_k, \ k \in \mathscr{N}_i^\Gamma\}.$$

By using above definitions, we have the following finite element discretization for the two-domains method:

$$\begin{cases} \text{find } \ z_1^h \in K_h^e \ \ \forall v_h \in K_h^e \\ a^e(z_1^h, v_h) = (f^e, v_h), \end{cases} \tag{9}$$

$$\begin{cases} \text{find } \lambda_h \in S^h \text{ and } w_h \in K_h^i \text{ such that } \forall v_h \in K_h^i, \ \forall \ \mu_h \in S^h, \\ (\mathscr{S}^e\lambda_h, \mu_h) = ((f^e, F^e\mu_h^e) - a^e(z_1^h, F^e\mu_h^e)) + ((f^i, F^i\mu_h^i) - a^i(w_h, F^i\mu_h^i)), \\ a^i(w_h, v_h - w_h) \geq (f^i, v_h - w_h) \end{cases} \tag{10}$$

$$\begin{cases} \text{find } z_2^h = (E\lambda)_h + K_h^e \text{ such that } \ \forall v_h \in K_h^e, \\ a^e(z_h, v_h) = 0. \end{cases} \tag{11}$$

2.2 Matrix formulation

To obtain the matrix formulation of the above discrete formulation of the domain decomposition problem let us denote the unknown vectors by $\mathbf{u}^e, \mathbf{u}^i, \mathbf{u}^\Gamma$ and the right hand side vectors by $\mathbf{f}^e, \mathbf{f}^i, \mathbf{f}_\Gamma$ of lengths N^e, N^i, N^Γ respectively, such that $N = N^e + N^i + N^\Gamma$, with $A \in \mathbb{R}^{N\times N}$ and $\mathbf{f} \in \mathbb{R}^N$. Then the matrix representation of (1) can be written as

$$\begin{cases} A\mathbf{u} \ \geq \mathbf{f}, \\ \mathbf{u} \ \geq \Psi, \end{cases}$$

subject to the complementarity conditions $(\mathbf{f} - A\mathbf{u})_j(\mathbf{u} - \Psi)_j = 0$, with

$$\begin{pmatrix} A_{II}^e & O & A_{I\Gamma}^e \\ O & A_{II}^i & A_{I\Gamma}^i \\ A_{\Gamma I}^e & A_{\Gamma I}^i & A_{\Gamma\Gamma} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \mathbf{u}_I^e \\ \mathbf{u}_I^i \\ \mathbf{u}_\Gamma \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \mathbf{f}_I^e \\ \mathbf{f}_I^i \\ \mathbf{f}_\Gamma \end{pmatrix}, \tag{12}$$

where we have partitioned the degrees of freedom into those internal to Ω^e and to Ω^i and those on the interface Γ. By using this notation, the above discrete weak formulations have the following matrix form

$$A_{II}^e \mathbf{u}_I^{e,1} = \mathbf{f}_I^e, \tag{13a}$$

$$S^e \mathbf{u}_\Gamma = \mathbf{f}_\Gamma - A_{\Gamma I}^e \mathbf{u}_I^{e,1} - A_{\Gamma I}^i \mathbf{u}_I^i, \tag{13b}$$

$$A_{II}^i \mathbf{u}_I^i \ \geq \ \mathbf{f}_I^i - A_{I\Gamma}^i \mathbf{u}_\Gamma, \tag{13c}$$

$$A_{II}^e \mathbf{u}_I^{e,2} = -A_{I\Gamma}^e \mathbf{u}_\Gamma, \tag{13d}$$

subject to conditions $(\mathbf{f}_I^i - A_{II}^i \mathbf{u}_I^i - A_{I\Gamma}^i \mathbf{u}_\Gamma)_j (\mathbf{u}_I^i - \Psi_I)_j = 0$, which represent the complementarity conditions for (13c).

The set of equations (13a)-(13d) could be seen as a partial Schur complement approach for the system (12). The solutions $\mathbf{u}_I^i$ and $\mathbf{u}_\Gamma$ will be approximated in an iterative manner. The resulting solution is then $[\mathbf{u}_I^{e,1} + \mathbf{u}_I^{e,2}, \mathbf{u}_I^i, \mathbf{u}_\Gamma]$.

2.3 Domain decomposition algorithm

Equations (13b) and (13c) form a coupled system which we solve by using a fixed point iteration. We note here that, given $\mathbf{u}_I^i$, the solution of (13b) involving the Schur complement matrix S^e can be implemented by using a Krylov subspace solver with domain decomposition preconditioning, corresponding to some partition of Ω^e into

several subdomains. On the other hand, (13c) is a standard linear complementarity problem posed on a small subdomain Ω^i. The proposed algorithm is included below.

Picard reduced QP algorithm

1: **step 0:** Find an initial guess by using coarse mesh solution
2: **step 1:** find $\mathbf{u}_I^{e\{1\}} = (A_{II}^e)^{-1}\mathbf{f}_I^e$,
3: **step 2:**
4: **for** $k = 0, 1, 2, \ldots$, till convergence **do**
5: Solve $S^e(\mathbf{u}_\Gamma)^{k+1} = (\mathbf{f}_\Gamma - A_{\Gamma I}^e\mathbf{u}_I^{e\{1\}} - A_{\Gamma I}^i(\mathbf{u}_I^i)^k)$
6: Find $(\mathbf{u}_I^i)^{k+1} \in K^i$ such that

$$J((\mathbf{u}_I^i)^{k+1}) \leq J(\mathbf{v}) \quad \forall \mathbf{v} \in \mathbf{K^i}$$

7: where

$$J(\mathbf{v}) := \frac{1}{2}(\mathbf{v})^T A_{II}^i \mathbf{v} - (\mathbf{v})^T(\mathbf{f}_I^i - A_{I\Gamma}^i \mathbf{u}_\Gamma^{k+1})$$

8: If converged, set $\mathbf{u}_\Gamma = \mathbf{u}_\Gamma^{k+1}$ and exit
9: **end for**
10: **step 3:** Compute

$$\mathbf{u}_I^{e\{2\}} = -(A_{II}^e)^{-1}A_{I\Gamma}^e\mathbf{u}_\Gamma$$

11: The resulting solution is then

$$\mathbf{u} = [\mathbf{u}_I^{e\{1\}} + \mathbf{u}_I^{e\{2\}}, \mathbf{u}_I^i, \mathbf{u}_\Gamma].$$

---

3 Numerical Experiments

Test 1:One obstacle

For our first test problem, we consider an elastic membrane which lies above an obstacle of height 1 centered at the origin with square cross-section with side length $\ell^o = 0.3$ under the forcing function $f = 1$ with $\Omega = (-1,1)^2$. We choose Ω^i to be a square region with side-length ℓ^i which contains the support of the obstacle such that the interface boundary Γ lies outside of the obstacle support. In the given algorithm we solved PDI, in the step 2(ii) by using the matlab function `quadprog`, a built-in quadratic programming solver. The PDI is coupled together with the interface equality problem in step 2(i) in an iterative manner. The relation to constrained minimization problems with quadratic programming problem can be found in [8]. We apply fixed point DD algorithm with global complementarity condition as a stopping criterion $\max_{1\leq i\leq n} \mid (Ł\mathbf{u} - \mathbf{f})_i(\mathbf{u} - \Psi)_i \mid \leq 10^{-3}$. The initial guess was computed on a fixed coarse mesh with n_0 nodes. Note that the variational inequality problem is now posed over a small subdomain, and hence has low complexity - we therefore decided not to report on it. Table 1 displays the number of fixed point iterations required to solve the coupled equations (13b), (13c). We see that the number

of iterations grows logarithmically as we increase the level of refinement. On the other hand, reducing the size of Ω^i leads to a smaller number of iterations, while preserving the dependence behaviour on the refinement level.

Table 1 Fixed point iterations for test problem 1.

ℓ^i=	0.4	0.5	0.6
n = 1,089	8	10	10
4,225	12	16	17
16,641	17	25	26

Test 2: Three obstacles

For the same domain Ω we consider the obstacle problem with three square obstacles of height 1 with centers located at $(0.5, 0.5), (-0.5, 0.5), (0, -0.5)$ and equal sides $\ell^o = 0.3$. We performed the same investigation, where we chose Ω^i to be a multiply-connected domain consisting of square regions of side-length ℓ^i (see Fig. 1. The numerical results are displayed in Table 2. For this harder problem, the number of iterations displays a logarithmic dependence for ℓ^i sufficiently small, but deteriorates for larger Ω^i. However, this is not the context we devised our algorithm for. Finally, we note that for this test problem the variational inequality in step (ii) decouples into three independent variational inequalities.

Table 2 Fixed point iterations for test problem 2.

ℓ^i=	0.4	0.5	0.6
n = 1,089	9	14	14
4,225	14	21	24
16,641	19	32	38

4 Summary and future work

We described an algorithm for the solution of obstacle problems using a two-domain formulation. In the larger subdomain we solved a PDE, while in the smaller region containing the coincidence set we solved a variational inequality using a minimization formulation. The solution of the PDE, as well as the solution involving a reduced Schur complement problem can in practice be achieved via a parallel implementation of a Krylov method coupled with a domain decomposition precondi-

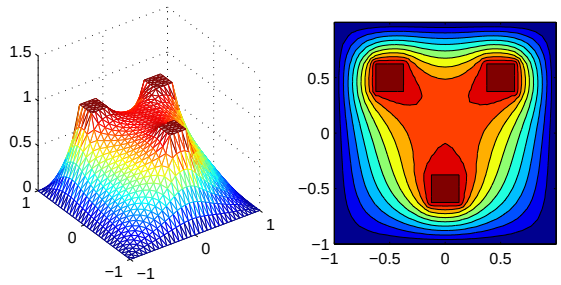


Fig. 1 Test problem 2: the choice of Ω^i for $\ell^i = 0.4$ and the corresponding solution.

tioner. Work in progress includes a Newton-Krylov solution of the non-linear problem (7). Future work is expected to include results validating this approach as well as an analysis of our algorithm. We are also interested to implement this method on general elliptic and parabolic problems.

References

1. Badea, L.: Multigrid methods for some quasi-variational inequalities. Discrete & Continuous Dynamical Systems-Series S, accepted for publication (2011)
2. Gelman, E., Mandel, J.: On multilevel iterative methods for optimization problems. Mathematical Programming **48**(1-3), 1–17 (1990)
3. Gräser, C., Kornhuber, R.: Multigrid methods for obstacle problems (2008)
4. Kornhuber, R.: Monotone multigrid methods for elliptic variational inequalities I. Numerische Mathematik **69**, 167–167 (1994)
5. Kornhuber, R., Yserentant, H.: Multilevel methods for elliptic problems on domains not resolved by the coarse grid. Contemporary Mathematics **180**, 49–49 (1994)
6. Mandel, J.: A multilevel iterative method for symmetric, positive definite linear complementarity problems. Applied Mathematics and Optimization **11**(1), 77–95 (1984)
7. Tai, X.: Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. Numerische Mathematik **93**(4), 755–786 (2003)
8. K.G. Murty:Linear Complementarity, Linear and Nonlinear Programming
9. Tai, X.C., Heimsund, B.o., Xu, J.: Rate of convergence for parallel subspace correction methods for nonlinear variational inequalities. In: Thirteenth International Domain Decomposition Conference, pp. 127–138 (2001)
10. Kinderlehrer, D.: Elliptic variational inequalities. In: Proc. Intl. Congress of Math. Vancouver, pp. 269–273 (1974)

A domain decomposition algorithm for contact problems with Coulomb's friction

J. Haslinger[1], R. Kučera[2], and T. Sassi[1]

1 Introduction

Contact problems of elasticity are used in many fields of science and engineering, especially in structural mechanics, geology and biomecanics. Many numerical procedures solving contact problems have been proposed in the engineering literature. They are based on standard discretization techniques for partial differential equations in combination with a special implementation of non-linear contact conditions (e.g., see [3, 5, 6, 8]).

The use of domain decomposition methods turns out to be one of the most efficient approaches. Recently, Dirichlet-Neumann and FETI type algorithms have been proposed and studied for solving multibody contact problems with Coulomb friction (see for example [7, 1, 2]).

In this paper, the Neumann-Neumann algorithm is extended to two-body contact problems with Coulomb friction. The main difficulty is due to the boundary conditions at the contact interface. They are highly non-linear, both in the normal direction (unilateral contact conditions) and in the tangential one (Coulomb's law). A fixed point procedure is introduced to ensure the continuity of the contact stresses. Numerical results illustrate that an optimal relaxation parameter exists and its value is nearly independent of the friction coefficient and the mesh size.

2 Setting of the problem

Let us consider two plane elastic bodies, occupying bounded domains Ω^α, $\alpha = 1,2$. The boundary $\Gamma^\alpha := \partial\Omega^\alpha$ is assumed to be piecewise continuous, and it is split into three non empty disjoint parts Γ_u^α, Γ_p^α and Γ_c^α such that $\overline{\Gamma_u^\alpha} \cap \overline{\Gamma_c^\alpha} = \emptyset$. Each body Ω^α is fixed on Γ_u^α and subject to surface tractions $\phi^\alpha \in (L^2(\Gamma_p^\alpha))^2$ on Γ_p^α. The body forces are denoted by $f^\alpha \in (L^2(\Omega^\alpha))^2$. In the initial configuration, both bodies have a common contact portion $\Gamma_c := \Gamma_c^1 = \Gamma_c^2$. In other words, we consider the case when the contact zone cannot grow during the deformation process and there is no gap between Ω^1 and Ω^2. Unilateral contact conditions with local Coulomb's friction are prescribed on Γ_c. The problem consists in finding the displacement field $u = (u^1, u^2)$

[1]KNM MFF UK Prague, Czech Republic, Sokolovská 83, 18675 Praha 8,[2]Department of Mathematics and Descriptive Geometry, VŠB-TUO, Czech Republic, 17. listopadu 15/2172, 70833 Ostrava-Poruba,[3]Laboratoire de Mathématiques Nicolas Oresme, Université de Caen Basse-Normandie, France, e-mail: {hasling@karlin.mff.cuni.cz}{radek.kucera@vsb.cz}{taoufik.sassi@unicaen.fr}

(the notation u^α stands for $u|_{\Omega^\alpha}$) and the stress tensor field $\sigma = (\sigma(u^1), \sigma(u^2))$ such that:

$$\left.\begin{array}{lll} div\,\sigma(u^\alpha) + f^\alpha = 0 & \text{in } \Omega^\alpha, \\ \sigma(u^\alpha)n^\alpha = \phi^\alpha & \text{on } \Gamma_p^\alpha, \\ u^\alpha = 0 & \text{on } \Gamma_u^\alpha, \end{array}\right\} \tag{1}$$

$\alpha = 1,2$. The elastic constitutive law, is given by Hooke's law for homogeneous and isotropic material:

$$\sigma_{ij}(u^\alpha) = \mathsf{A}^\alpha_{ijkh} e_{kh}(u^\alpha), \quad e(u^\alpha) = \frac{1}{2}\left(\nabla u^\alpha + (\nabla u^\alpha)^T\right), \tag{2}$$

where $\mathsf{A}^\alpha = (A^\alpha_{ijkh})_{1\le i,j,k,h\le 2} \in (L^\infty(\Omega^\alpha))^{16}$ is the fourth-order elasticity tensor satisfying the usual symmetry and ellipticity conditions and $e(u^\alpha)$ is the respective strain tensor. The summation convention is adopted.

Further the normal and tangential components of the displacement u and the stress vector on Γ_c are defined by

$$\left.\begin{array}{ll} u_N^\alpha = u_i^\alpha n_i^\alpha, & u_{T_i}^\alpha = u_i^\alpha - u_N^\alpha n_i^\alpha, \\ \sigma_N^\alpha = \sigma_{ij}(u^\alpha) n_i^\alpha n_j^\alpha, & {\sigma_T}_i^\alpha = \sigma_{ij}(u^\alpha) n_j^\alpha - \sigma_N^\alpha n_i^\alpha, \end{array}\right\} \tag{3}$$

where n^α denotes the outward normal unit vector to the boundary. On the interface Γ_c, the unilateral contact law conditions are prescribed:

$$\sigma_N := \sigma_N^1 = \sigma_N^2, \qquad \sigma_T := \sigma_T^1 = \sigma_T^2, \tag{4}$$

$$[u_N] \le 0, \quad \sigma_N \le 0, \quad \sigma_N [u_N] = 0, \tag{5}$$

where $[v_N] = v^1 \cdot n^1 + v^2 \cdot n^2$ is the jump across the interface Γ_c of a function v defined on $\Omega^1 \cup \Omega^2$. Coulomb's law of local friction reads as follows

$$\left.\begin{array}{l} |\sigma_T| \le \mathscr{F}|\sigma_N|, \\ |\sigma_T| < \mathscr{F}|\sigma_N| \Longrightarrow [u_T] = 0, \\ |\sigma_T| = \mathscr{F}|\sigma_N| \Longrightarrow \exists \nu \ge 0 \quad [u_T] = -\nu\sigma_T, \end{array}\right\} \tag{6}$$

where $\mathscr{F} \in L^\infty(\Gamma_c)$, $\mathscr{F} \ge 0$ on Γ_c is the coefficient of friction and $[u_T]$ stands for the jump of the tangential displacements.

Weak solutions of the contact problem obeying Coulomb's law of friction can be defined as a fixed point of the mapping $\Phi : \Lambda \mapsto \Lambda$, where $\Lambda = \{\mu \in H^{-1/2}(\Gamma_c), \mu \ge 0\}$ and $\Phi(g) = -\sigma_N(u)$ with $u \in \mathbb{K}$ being the unique solution of the variational inequality:

$$u := u(g) \in \mathbb{K}: \; a(u, v-u) + \langle \mathscr{F} g, |[v_T]| - |[u_T]| \rangle \ge L(v-u), \quad \forall v \in \mathbb{K}. \qquad (\mathscr{P})$$

Here

$$\mathbb{K} = \{v \in \mathbb{V} |\ [v_N] \le 0 \text{ on } \Gamma_c\}, \qquad \mathbb{V} = \mathbb{V}^1 \times \mathbb{V}^2,$$
$$\mathbb{V}^\alpha = \{v^\alpha \in (H^1(\Omega^\alpha))^2 \mid v^\alpha = 0 \ \text{ on } \Gamma_u^\alpha\},\ \alpha = 1,2.$$

The bilinear and linear form $a(\cdot,\cdot)$, $L(\cdot)$ represent the inner energy of the system, and the work of applied forces, respectively:

$$a(v,w) = a^1(v^1,w^1) + a^2(v^2,w^2), \qquad L(v) = L^1(v^1) + L^2(v^2),\ \ v,w \in \mathbb{V},$$

where

$$a^\alpha(v^\alpha,w^\alpha) = \int_{\Omega^\alpha} \mathsf{A}^\alpha_{ijkh} e_{kh}(v^\alpha) e_{ij}(w^\alpha)\,dx,$$
$$L^\alpha(v^\alpha) = \int_{\Omega^\alpha} f^\alpha \cdot v^\alpha\,dx + \int_{\Gamma_p^\alpha} \phi^\alpha \cdot v^\alpha\,ds,$$

$\alpha = 1,2$. The symbol $\langle\cdot,\cdot\rangle$ stands for the duality pairing between $H^{-1/2}(\Gamma_c)$ and $H^{1/2}(\Gamma_c)$ or for the scalar product in $L^2(\Gamma_c)$, if $g \in L^2(\Gamma_c)$.

3 Domain decomposition algorithm for contact problems with given friction

We present the continuous version of the domain decomposition algorithm for solving $(\mathscr{P})$. The mathematical justification of all results presented below can be found in [4]. We introduce the following notation: by $\pi^\alpha : (H^{1/2}(\Gamma_c))^2 \mapsto \mathbb{V}^\alpha$ we denote the extension mapping defined for $\lambda \in (H^{1/2}(\Gamma_c))^2$ by

$$\left.\begin{array}{c} \pi^\alpha\lambda \in \mathbb{V}^\alpha : a^\alpha(\pi^\alpha\lambda, v^\alpha) = 0 \quad \forall v^\alpha \in \mathbb{V}_0^\alpha, \\ \pi^\alpha\lambda = \lambda \ \text{ on } \ \Gamma_c, \end{array}\right\} \tag{7}$$

where

$$\mathbb{V}_0^\alpha = \{v^\alpha \in (H^1(\Omega^\alpha))^2 |\ v^\alpha = 0 \text{ on } \ \Gamma_u^\alpha \cup \Gamma_c\}. \tag{8}$$

Further for $\varphi \in L^2(\Gamma_c)$ given, we define:

$$\mathbb{K}^2(\varphi) = \{v^2 \in \mathbb{V}^2 |\ v^2 \cdot n^2 \le -\varphi \text{ on } \Gamma_c\}$$

and the frictional term $j : \mathbb{V} \mapsto \mathbb{R}$ by

$$j(v) := j(v_1,v_2) = \int_{\Gamma_c} g|[v_T]|\,ds, \quad v = (v_1,v_2) \in \mathbb{V}.$$

The algorithm is based on the following result.

Proposition 1. *A pair $u=(u^1,u^2)\in\mathbb{V}$ is a solution of $(\mathscr{P})$ if and only if $u^1\in\mathbb{V}^1$, $u^2\in\mathbb{V}^2$ solve the following problems:*

$$\left.\begin{aligned}&\text{Find } u^1\in\mathbb{V}^1 \text{ such that}\\ &a^1(u^1,v^1)=L^1(v^1)-a^2(u^2,\pi^2v^1)+L^2(\pi^2v^1)\quad\forall v^1\in\mathbb{V}^1\end{aligned}\right\} \tag{9}$$

and

$$\left.\begin{aligned}&\text{Find } u^2\in\mathbb{K}^2(u^1\cdot v^1) \text{ such that}\\ &a^2(u^2,v^2-u^2)+j(u^1,v^2)-j(u^1,u^2)\geq L^2(v^2-u^2)\quad\forall v^2\in\mathbb{K}^2(u^1\cdot v^1),\end{aligned}\right\} \tag{10}$$

respectively.

Suppose that $\lambda\in(H^{1/2}(\Gamma_c))^2$ is given and u^1, u^2 are the solutions of the following *decoupled* problems:

$$\left.\begin{aligned}&\text{Find } u^1:=u^1(\lambda)\in\mathbb{V}^1 \text{ such that}\\ &a^1(u^1,v^1)=L^1(v^1)\quad\forall v^1\in\mathbb{V}^1_0\\ &u^1=\lambda\quad\text{on }\Gamma_c\end{aligned}\right\} \tag{$\mathscr{P}_1(\lambda)$}$$

and

$$\left.\begin{aligned}&\text{Find } u^2:=u^2(\lambda)\in\mathbb{K}^2(\lambda\cdot n^1) \text{ such that}\\ &a^2(u^2,v^2-u^2)+j(\lambda,v^2)-j(\lambda,u^2)\geq L^2(v^2-u^2)\\ &\qquad\qquad\forall v^2\in\mathbb{K}^2(\lambda\cdot n^1).\end{aligned}\right\} \tag{$\mathscr{P}_2(\lambda)$}$$

If $\lambda\in(H^{1/2}(\Gamma_c))^2$ was chosen in such a way that $\sigma_N^1=\sigma_N^2$ and $\sigma_T^1=\sigma_T^2$ on Γ_c, then the couple $u=(u^1,u^2)\in\mathbb{K}$ would be a solution of $(\mathscr{P})$. To find such λ ensuring continuity of the normal and tangential contact stress across Γ_c, we shall use the following auxiliary Neumann problems defined in Ω^1 and Ω^2:

$$\left.\begin{aligned}&\text{Find } w^1\in\mathbb{V}^1 \text{ such that}\\ &a^1(w^1,v^1)=\tfrac{1}{2}(-a^1(u^1,v^1)+L^1(v^1)-a^2(u^2,\pi^2v^1)+L^2(\pi^2v^1))\\ &\qquad\qquad\forall v^1\in\mathbb{V}^1\end{aligned}\right\} \tag{$\mathscr{P}_3(\lambda)$}$$

and

$$\left.\begin{aligned}&\text{Find } w^2\in\mathbb{V}^2 \text{ such that}\\ &a^2(w^2,v^2)=\tfrac{1}{2}(a^2(u^2,v^2)-L^2(v^2)+a^1(u^1,\pi^1v^2)-L^1(\pi^1v^2))\\ &\qquad\qquad\forall v^2\in\mathbb{V}^2,\end{aligned}\right\} \tag{$\mathscr{P}_4(\lambda)$}$$

where $u^1:=u^1(\lambda)$, $u^2:=u^2(\lambda)$ are the solutions of $(\mathscr{P}_1(\lambda))$, and $(\mathscr{P}_2(\lambda))$, respectively. The algorithm consists of the following five steps:

ALGORITHM (**DD**) Let $\lambda_0 \in (H^{1/2}(\Gamma_c))^2$ and $\theta > 0$ be given. For $k \geq 1$ integer, define u_k^α, w_k^α, $\alpha = 1,2$ and λ_k by:

Step 1. $u_k^1 \in \mathbb{V}^1$ solves $(\mathscr{P}_1(\lambda_{k-1}))$;

Step 2. $u_k^2 \in \mathbb{K}^2(\lambda_{k-1} \cdot n^1)$ solves $(\mathscr{P}_2(\lambda_{k-1}))$;

Step 3. $w_k^1 \in \mathbb{V}^1$ solves $(\mathscr{P}_3(\lambda_{k-1}))$;

Step 4. $w_k^2 \in \mathbb{V}^2$ solves $(\mathscr{P}_4(\lambda_{k-1}))$;

Step 5. $\lambda_k = \lambda_{k-1} + \theta(w_k^1 - w_k^2)$ on Γ_c.

The convergence property of this algorithm follows from the next theorem.

Theorem 1. *There exist:* $0 < \theta^* < 4$ *and functions* $\lambda_* \in (H^{1/2}(\Gamma_c))^2$, $u_*^\alpha, w_*^\alpha \in \mathbb{V}^\alpha$, $\alpha = 1,2$ *such that for any* $\theta \in (0, \theta^*)$ *it holds:*

$$\left.\begin{array}{ll} \lambda_k \to \lambda_* & \text{in } (H^{1/2}(\Gamma_c))^2, \\ u_k^\alpha \to u_*^\alpha \\ w_k^\alpha \to w_*^\alpha \end{array}\right\} \text{ in } (H^1(\Omega^\alpha))^2,\ \alpha = 1,2, \Bigg\} \quad k \to \infty \tag{11}$$

where the sequence $\{(u_k^\alpha, w_k^\alpha, \lambda_k)\}$ *is generated by* ALGORITHM (**DD**). *In addition, the couple* (u_*^1, u_*^2) *solves* $(\mathscr{P})$.

A discrete version of algorithm is obtained by a finite element approximation of *Steps 1-4*. In [4] we used piecewise linear functions on triangulations of Ω^1 and Ω^2. These triangulations are supposed to be compatible on the contact part Γ_c. Using a similar technique as in Theorem 1, one can prove the convergence property of the discrete version with θ^* independent of the mesh norm.

4 Numerical experiments

In this section, we shall test the performance of variants of ALGORITHM (**DD**) for solving contact problems with Coulomb friction. For this reason, we combine ALGORITHM (**DD**) with the method of succesive approximations that enables us to compute fixed points of the mapping Φ. To get an efficient algorithm, we perform only one iteration of ALGORITHM (**DD**) in each step of the method of succesive approximations. In other words, we update the slip bound g in each *Step 2* using the result of the previous iteration, i.e., $g = -\sigma_N(u_{k-1}^2)$ (and $g = 0$, if $k = 1$). This algorithm will be called ALGORITHM I in this numerical part.

Note that *Step 2* in ALGORITHM I treats simultaneously both, the non-penetration and the friction conditions. A natural idea occurs, namely to split these conditions between *Steps 1* and *2*. This modification of ALGORITHM (**DD**) will be called ALGORITHM II.

In both, ALGORITHM I and II, one can perform splitting of the Gauss-Seidel type so that computatios of the normal and tangential contact stresses are decoupled by performing one Gauss-Seidel iteration; see [4] for more details. In the respective columns of the tables below we show the results without (column *without*) and with the Gauss-Seidel splitting in *Step 1*, *2*, and in both these steps.

Example 1. Let us consider two plane elastic bodies

$$\Omega^1 = (0,3)\times(1,2) \quad \text{and} \quad \Omega^2 = (0,3)\times(0,1)$$

made of an isotropic, homogeneous material characterized by the Young modulus 2.1×10^{11} and the Poisson ratio 0.277 (steel). The decompositions of $\partial\Omega^\alpha$, $\alpha = 1,2$ are as follows:

$$\Gamma_u^1 = \{0\}\times(1,2),\ \Gamma_c^1 = (0,3)\times\{1\},\ \Gamma_p^1 = \partial\Omega^1 \setminus \overline{\Gamma_u^1 \cup \Gamma_c^1},$$
$$\Gamma_u^2 = \{0\}\times(0,1),\ \Gamma_c^2 = (0,3)\times\{1\},\ \Gamma_p^2 = \partial\Omega^2 \setminus \overline{\Gamma_u^2 \cup \Gamma_c^2}.$$

The volume forces $f^\alpha = 0$ in Ω^α, $\alpha = 1,2$ while the following surface tractions of density $\phi^1 = (\phi_1^1, \phi_2^1)$ act on Γ_p^1:

$$\phi_1^1(s,2) = 0,\ \phi_2^1(s,2) = \phi_{2,L}^1 + \phi_{2,R}^1 s, \quad s\in(0,3),$$
$$\phi_1^1(3,s) = \phi_{1,B}^1(2-s) + \phi_{1,U}^1(s-1), \quad s\in(1,2),$$
$$\phi_2^1(3,s) = \phi_{2,B}^1(2-s) + \phi_{2,U}^1(s-1), \quad s\in(1,2),$$

where $\phi_{2,L}^1 = -6\times 10^7$, $\phi_{2,R}^1 = -1/3\times 10^7$, $\phi_{1,B}^1 = 2\times 10^7$, $\phi_{1,U}^1 = 2\times 10^7$, $\phi_{2,B}^1 = 4\times 10^7$, and $\phi_{2,U}^1 = 2\times 10^7$. The coefficient of friction is $\mathscr{F} = 0.3$.

We compare performance of ALGORITHMS I and II with different splittings of Gauss-Seidel type for various values of θ and degrees of freedom n (twice the number of nodes) and m (the number of the contact nodes). In the tables we report the computational time in seconds, the number $\#iter$ of the (outer) iterations, and the total number of actions $n_{\mathbf{A}}$ of the inverses to the stiffness matrices. Further we quote the total efficiency of the method assessed by the ratio $eff := n_{\mathbf{A}}/(2m)$ which gives a comparison of our algorithms with the realization of "similar linear problems" by the standard conjugate gradient method. It is well-known that the number of conjugate gradient iterations, i.e. the number of matrix-vector multiplications, is bounded by the size of the problem. Therefore, one can say that our algorithms exhibit the complexity comparable with the conjugate gradient method when *eff* is less than two. All computations are performed in Matlab 8.2 on Intel(R)Core(TM)2 Duo CPU, 2 GHz with 3 GB RAM. We set the relative terminating precision on the computed contact stresses to $tol = 10^{-4}$. The inner problems in *Step 1* and *2* are solved by optimization algorithms based on the conjugate gradient method with the adaptive precision control respecting the accuracy achieved in the outer loop; see [4] for more details.

Table 1 Characteristics of ALGORITHM I without and with splitting.

$n\|m$	*without* $\#iter\|n_{\mathbf{A}}$ $[time\|eff]$	*in Step 1* $\#iter\|n_{\mathbf{A}}$ $[time\|eff]$	*in Step 2* $\#iter\|n_{\mathbf{A}}$ $[time\|eff]$	*in Step 1+2* $\#iter\|n_{\mathbf{A}}$ $[time\|eff]$
504|18	60|667 [0.80|18.53]	61|1075 [0.98|29.86]	59|742 [0.67|20.61]	60|1146 [0.70|31.83]
6072|66	61|1044 [8.19|7.91]	61|1492 [8.35|11.30]	61|824 [4.63|6.24]	60|1236 [6.91|9.36]
17784|114	62|1313 [31.73|5.76]	63|1816 [43.71|7.96]	61|855 [33.24|3.75]	63|1365 [32.89|5.99]
35640|162	61|1839 [126.94|5.68]	62|1819 [133.30|5.61]	61|892 [59.59|2.75]	62|1377 [91.82|4.25]
59640|210	60|1583 [238.32|3.77]	61|2336 [341.33|5.56]	61|876 [127.42|2.09]	61|1377 [196.11|3.28]
89784|258	60|1627 [405.31|3.15]	59|2333 [585.25|4.52]	60|864 [216.09|1.67]	61|1421 [359.08|2.75]

Table 2 Characteristics of ALGORITHM II without and with splitting.

$n\|m$	*without* $\#iter\|n_{\mathbf{A}}$ $[time\|eff]$	*in Step 1* $\#iter\|n_{\mathbf{A}}$ $[time\|eff]$	*in Step 2* $\#iter\|n_{\mathbf{A}}$ $[time\|eff]$	*in Step 1+2* $\#iter\|n_{\mathbf{A}}$ $[time\|eff]$
504|18	37|530 [0.19|14.72]	36|520 [0.16|14.44]	37|714 [0.19|19.83]	38|770 [0.19|21.39]
6072|66	36|987 [5.76|7.48]	37|586 [3.29|4.44]	37|964 [5.35|7.30]	38|829 [4.59|6.28]
17784|114	36|1417 [34.32|6.21]	38|626 [15.16|2.75]	37|1347 [32.81|5.91]	35|794 [19.00|3.48]
35640|162	37|1864 [119.50|5.75]	36|608 [38.74|1.88]	36|1399 [89.79|4.32]	36|863 [54.83|2.66]
59640|210	37|2132 [290.71|5.08]	37|624 [93.40|1.49]	37|1401 [191.30|3.34]	35|851 [115.64|2.03]
89784|258	37|2532 [631.80|4.91]	37|619 [154.52|1.20]	37|1806 [451.65|3.50]	36|877 [225.59|1.70]

Figure 1 illustrates the sensitivity of the different variants of our algorithms with respect to θ. From these results one may conclude at least two facts: *(i)* ALGORITHM II without splitting is more stable than ALGORITHM I in sense that it converges for larger values of θ; *(ii)* splitting used *Step 2* of ALGORITHM II leads to the convergent process for all $\theta \in (0,1]$.

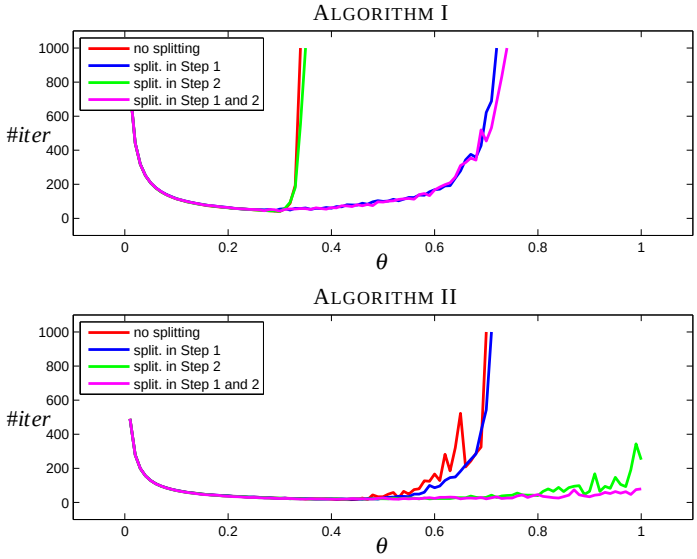


Fig. 1 For each θ we display the number of iterations $\#iter$ satisfying the terminating precision as above ($n = 1872$, $m = 36$).

Acknowledgements This research was supported by the grant GAČR P201/12/0671.

References

1. Bayada, G., Sabil, J., Sassi, T.: Convergence of neumann-dirichlet algorithm for two-body contact problems with non local Coulomb's friction law. ESAIM: Mathematical Modelling and Numerical Analysis **42**(4), 243–262 (2008)
2. Dostál, Z., Kozubek, T., Horyl, P., T. Brzobohatý, A.: Scalable TFETI algorithm for two dimensional multibody contact problems with friction. Journal of Computational and Applied Mathematics **235**, 403–418 (2010)
3. Haslinger, J., Dostál, Z., Kučera, R.: On a splitting type algorithm for the numerical realization of contact problems with Coulomb friction. Computer Methods in Applied Mechanics and Engineering **191**(21-22), 2261–2281 (2002)
4. Haslinger, J., Kučera, R., Riton, J., Sassi, T.: A domain decomposition method for two-body contact problems with Tresca friction. Advances in Computational Mathematics (acceptted) (2012)
5. Kikuchi, N., Oden, J.T.: Contact problems in elasticity: a study of variational inequalities and finite element methods, *SIAM Studies in Applied Mathematics*, vol. 8. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1988)
6. Kornhuber, R., Krause, R.: Adaptive multigrid methods for Signorini's problem in linear elasticity. Computing and Visualization in Science **4**(1), 9–20 (2001)
7. Krause, R., Wohlmuth, B.: A Dirichlet-Neumann type algorithm for contact problems with friction. Computing and Visualization in Science **5**(3), 139–148 (2002)
8. Wriggers, P.: Computational Contact Mechanics. Springer, Berlin Heidelberg (2006)

Hybrid dual-primal FETI-Schur complement method for Stokes

Ange B. Toulougoussou[1] and François-Xavier Roux[2]

1 Discrete Stokes

The algebraic Stokes is of the following saddle point form: Find $(U_h, P_h) \in V_h \times Q_h$ such that

$$\begin{pmatrix} A_h & B_h^T \\ B_h & 0 \end{pmatrix} \begin{pmatrix} U_h \\ P_h \end{pmatrix} = \begin{pmatrix} F_h \\ 0 \end{pmatrix}. \tag{1}$$

We suppose that the system (1) arises from the mixed finite-element discretization of Stokes on a domain Ω. We consider spaces V_h and Q_h that satisfy the inf-sup condition and whose elements are continuous. Such spaces can be found in [5] and include Hood-Taylor and Mini elements. Under the inf-sup condition and assuming a mixed boundary condition on the velocity there exists a unique solutuion to (1).

2 Hybrid dual-primal FETI-Schur

Stokes is a bottleneck in the analysis of incompressible fluid flows and is the subject of many researches. The numerical solution of the system (1) that arises from its discretization is a challenging problem because of the indefiniteness of saddle-point problems [1]. Memory space storage is an other important issue to deal with for large three-dimensional problems. An overview of solution methods to solve saddle-point problems is given in [1]. We focus on iterative methods such as FETI and BDD that save memory space and have proved efficiency for many linear systems. The domain Ω is split into N non-overlapping subdomains $\{\Omega^{(s)}\}_{s=1,\cdots,N}$ with interface $\Gamma_I = \cup_{s,q=1}^{N} \{\overline{\Omega}^{(s)} \cap \overline{\Omega}^{(q)}\}$. Degrees of freedom of each subdomain $\Omega^{(s)}$ are split into internal degrees of freedom designated by subscript i and degrees of freedom designated by subscript Γ that correspond to the interface of the subdomain $\Omega^{(s)}$ with other subdomains. Related to the spliting above, FETI and BDD split the original linear systems into subproblems whose solutions are flux and trace continuous respectively [4, 9]. FETI adresses these compatibility requirements by introducing a unique Lagrange multiplier on the interface to ensure the weak continuity of the subsolutions. FETI is dual to BDD that imposes a unique trace to the subsolutions on the interface. The original system is thus reduced in both cases to interface problems to be solved by Krylov methods that nullify the residual at convergence. The resid-

[1] Université Pierre et Marie Curie, 4 place Jussieu 75005 Paris, France e-mail: toulougoussou@ann.jussieu.fr .[2] ONERA, Chemin de la Hunière et des Joncherettes, BP 80100,FR-91123 PALAISEAU CEDEX e-mail: roux@onera.fr

uals in FETI and BDD are the jump of the solutions and of the flux on the interface respectively.These domain decomposition methods have been successfully extended to solve the system (1) when the discrete pressure is discontinuous. Their interface systems become mixed problems when the discrete velocity and pressure are both continuous. The spectral distribution of the interface operators slows down the rate of convergence of FETI and BDD that is proven to be optimal for systems arising from the discretization of elliptic problems. The interface unknowns resulting from the combination of FETI and BDD should be physically homogeneous [6] and well-suited for saddle-point problems such as (1) that arise in many applications [1]. We split the system (1) into N subsystems, renumber the unkwnows starting with the internal ones to get the following system:
Local systems

$$\begin{pmatrix} A_{ii}^{(s)} & {B_{ii}^{(s)}}^T \\ B_{ii}^{(s)} & 0 \end{pmatrix} \begin{pmatrix} U_i^{(s)} \\ P_i^{(s)} \end{pmatrix} + \begin{pmatrix} A_{i\Gamma} \\ B_{i\Gamma} \end{pmatrix} U_\Gamma^{(s)} + \begin{pmatrix} {B_{i\Gamma}^{(s)}}^T \\ 0 \end{pmatrix} P_\Gamma^{(s)} = \begin{pmatrix} F_i^{(s)} \\ 0 \end{pmatrix}, \tag{2}$$

interface problems

$$\begin{pmatrix} A_{\Gamma i}^{(s)} & {B_{i\Gamma}^{(s)}}^T \end{pmatrix} \begin{pmatrix} U_i^{(s)} \\ P_i^{(s)} \end{pmatrix} + A_{\Gamma\Gamma}^{(s)} U_\Gamma^{(s)} + {B_{\Gamma\Gamma}^{(s)}}^T P_\Gamma^{(s)} = F_\Gamma^{(s)}, \tag{3}$$

incompressibility conditions

$$\begin{pmatrix} B_{\Gamma i}^{(s)} & 0 \end{pmatrix} \begin{pmatrix} U_i^{(s)} \\ P_i^{(s)} \end{pmatrix} + B_{\Gamma\Gamma}^{(s)} U_\Gamma^{(s)} = 0, \quad s = 1, \cdots, N. \tag{4}$$

Systems (2)-(4) supplemented with continuity conditions on the velocity and on the pressure through the interface are equivalent to system (1).
Introduce notations:

$$\begin{aligned}
M_{uu}^{(s)} &= A_{\Gamma\Gamma}^{(s)} - \begin{pmatrix} A_{\Gamma i}^{(s)} & {B_{i\Gamma}^{(s)}}^T \end{pmatrix} \begin{pmatrix} A_{ii}^{(s)} & {B_{ii}^{(s)}}^T \\ B_{ii}^{(s)} & 0 \end{pmatrix}^{-1} \begin{pmatrix} A_{i\Gamma}^{(s)} \\ B_{i\Gamma}^{(s)} \end{pmatrix}, \\
M_{up}^{(s)} &= {B_{\Gamma\Gamma}^{(s)}}^T - \begin{pmatrix} A_{\Gamma i}^{(s)} & {B_{i\Gamma}^{(s)}}^T \end{pmatrix} \begin{pmatrix} A_{ii}^{(s)} & {B_{\Gamma\Gamma}^{(s)}}^T \\ B_{ii}^{(s)} & 0 \end{pmatrix}^{-1} \begin{pmatrix} {B_{i\Gamma}^{(s)}}^T \\ 0 \end{pmatrix}, \\
M_{pu}^{(s)} &= B_{\Gamma\Gamma}^{(s)} - \begin{pmatrix} B_{\Gamma i}^{(s)} & 0 \end{pmatrix} \begin{pmatrix} A_{ii}^{(s)} & {B_{ii}^{(s)}}^T \\ B_{ii}^{(s)} & 0 \end{pmatrix}^{-1} \begin{pmatrix} A_{i\Gamma}^{(s)} \\ B_{i\Gamma}^{(s)} \end{pmatrix}, \\
M_{pp}^{(s)} &= \begin{pmatrix} B_{\Gamma i}^{(s)} & 0 \end{pmatrix} \begin{pmatrix} A_{ii}^{(s)} & {B_{ii}^{(s)}}^T \\ B_{ii}^{(s)} & 0 \end{pmatrix}^{-1} \begin{pmatrix} {B_{\Gamma i}^{(s)}}^T \\ 0 \end{pmatrix}, \\
\tilde{F}_\Gamma^{(s)} &= F_\Gamma^{(s)} - \begin{pmatrix} A_{\Gamma i}^{(s)} & {B_{i\Gamma}^{(s)}}^T \end{pmatrix} \begin{pmatrix} A_{ii}^{(s)} & {B_{ii}^{(s)}}^T \\ B_{ii}^{(s)} & 0 \end{pmatrix}^{-1} \begin{pmatrix} F_i^{(s)} \\ 0 \end{pmatrix},
\end{aligned}$$

$$\tilde{F}_i^{(s)} = \begin{pmatrix} B_{\Gamma i}^{(s)} & 0 \end{pmatrix} \begin{pmatrix} A_{ii}^{(s)} & B_{ii}^{(s)^T} \\ B_{ii}^{(s)} & 0 \end{pmatrix}^{-1} \begin{pmatrix} F_i^{(s)} \\ 0 \end{pmatrix}, \quad s = 1, \cdots, N.$$

Lemma 1. *The subdomain Schur complements $M_{pp}^{(s)}$ and $M_{uu}^{(s)}$ are symmetric, positive semi-definite.*

Proof. Matrices $M_{pp}^{(s)}$ are clearly symmetric. Systems (2) are well-posed algebraic problems although they are not the usual Stokes because of the Dirichlet boundary condition on the pressure [2]. Therefore, for any given $P_\Gamma^{(s)}$, there exists

$$\begin{pmatrix} U_i^{(s)} \\ P_i^{(s)} \end{pmatrix} = - \begin{pmatrix} A_{ii}^{(s)} & B_{ii}^{(s)^T} \\ B_{ii}^{(s)} & 0 \end{pmatrix}^{-1} \begin{pmatrix} B_{i\Gamma}^{(s)^T} \\ 0 \end{pmatrix} P_\Gamma^{(s)}.$$

By Gaussian elimination, we have

$$\begin{pmatrix} A_{ii}^{(s)} & B_{ii}^{(s)^T} & B_{\Gamma i}^{(s)^T} \\ B_{ii}^{(s)} & 0 & 0 \\ B_{\Gamma i}^{(s)} & 0 & 0 \end{pmatrix} \begin{pmatrix} U_i^{(s)} \\ P_i^{(s)} \\ P_\Gamma^{(s)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -M_{pp}^{(s)} P_\Gamma^{(s)} \end{pmatrix}. \tag{5}$$

Therefore,

$$\begin{aligned} -P_\Gamma^{(s)^T} M_{pp}^{(s)} P_\Gamma^{(s)} &= \begin{pmatrix} U_i^{(s)} \\ P_i^{(s)} \\ P_\Gamma^{(s)} \end{pmatrix}^T \begin{pmatrix} A_{ii}^{(s)} & B_{ii}^{(s)^T} & B_{\Gamma i}^{(s)^T} \\ B_{ii}^{(s)} & 0 & 0 \\ B_{\Gamma i}^{(s)} & 0 & 0 \end{pmatrix} \begin{pmatrix} U_i^{(s)} \\ P_i^{(s)} \\ P_\Gamma^{(s)} \end{pmatrix} \\ &= U_i^{(s)^T} A_{ii}^{(s)} U_i^{(s)} + 2 P_i^{(s)^T} B_{ii}^{(s)} U_i^{(s)} + 2 P_\Gamma^{(s)^T} B_{\Gamma i}^{(s)} U_i^{(s)}. \end{aligned} \tag{6}$$

From (5), we have

$$B_{ii}^{(s)} U_i^{(s)} = 0 \quad \text{and} \quad B_{\Gamma i}^{(s)} U_i^{(s)} = -M_{pp}^{(s)} P_\Gamma^{(s)}.$$

Then from (6) and the positivity of the matrix arising from the discretization of the Laplace operator by finite elements, we have

$$P_\Gamma^{(s)^T} M_{pp}^{(s)} P_\Gamma^{(s)} = U_i^{(s)^T} A_{ii}^{(s)} U_i^{(s)} \geq 0.$$

We also have

$$\begin{pmatrix} A_{ii}^{(s)} & B_{ii}^{(s)^T} & B_{\Gamma i}^{(s)^T} \\ B_{ii}^{(s)} & 0 & 0 \\ B_{\Gamma i}^{(s)} & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1_i^{(s)} \\ 1_\Gamma^{(s)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{7}$$

where $1_i^{(s)}$ and $1_\Gamma^{(s)}$ are constants in the subdomain $\Omega^{(s)}$ and on its boundary respectively. By equality (7) one can show that in general there exists $R_p^{(s)}$ such that

$M_{pp}^{(s)} R_p^{(s)} = 0.$

It is well-known that the subdomain Schur complements $M_{uu}^{(s)}$ are symmetric, positive semi-definite in general [7].

Eliminating the internal degrees of freedom from local systems (2), the interface systems (3) and the incompressibility conditions (4) can be written as

$$\begin{pmatrix} M_{uu}^{(1)} & M_{up}^{(1)} & 0 & 0 & \cdots & \cdots & 0 & 0 \\ M_{pu}^{(1)} & -M_{pp}^{(1)} & 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & & \vdots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 0 & M_{uu}^{(N)} & M_{up}^{(N)} \\ 0 & 0 & \cdots & \cdots & 0 & 0 & M_{pu}^{(N)} & -M_{pp}^{(N)} \end{pmatrix} \begin{pmatrix} U_\Gamma^{(1)} \\ P_\Gamma^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ U_\Gamma^{(N)} \\ P_\Gamma^{(N)} \end{pmatrix} = \begin{pmatrix} \tilde{F}_\Gamma^{(1)} \\ -\tilde{F}_i^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \tilde{F}_\Gamma^{(N)} \\ -\tilde{F}_i^{(N)} \end{pmatrix}. \quad (8)$$

We introduce a unique Lagrange multiplier λ to ensure the weak continuity of the velocity on the interface as in FETI transforming the system (8) into

$$\begin{pmatrix} M_{uu}^{(1)} & M_{up}^{(1)} & 0 & 0 & \cdots & \cdots & 0 & 0 & T^{(1)^T} \\ M_{pu}^{(1)} & -M_{pp}^{(1)} & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & & \vdots & \vdots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 & 0 & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \ddots & 0 & 0 & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 0 & M_{uu}^{(N)} & M_{up}^{(N)} & T^{(N)^T} \\ 0 & 0 & \cdots & \cdots & 0 & 0 & M_{pu}^{(N)} & -M_{pp}^{(N)} & 0 \\ T^{(1)} & 0 & \cdots & \cdots & \cdots & \cdots & T^{(N)} & 0 & 0 \end{pmatrix} \begin{pmatrix} U_\Gamma^{(1)} \\ P_\Gamma^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ U_\Gamma^{(N)} \\ P_\Gamma^{(N)} \\ \lambda \end{pmatrix} = \begin{pmatrix} \tilde{F}_\Gamma^{(1)} \\ -\tilde{F}_i^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \tilde{F}_\Gamma^{(N)} \\ -\tilde{F}_i^{(N)} \\ 0 \end{pmatrix}. \quad (9)$$

where $\{T^{(s)}\}_{s=1,N}$ are boolean matrices of elements -1, 0 and 1. The application of the matrix $T^{(s)}$ to a matrix or a vector extracts and signs the interface components of that matrix or vector [4]. We next introduce the 0–1 matrix $L^{(s)^T}$ that maps the interface degrees of freedom of subdomain $\Omega^{(s)}$ into global interface degrees of freedom belonging to the interface Γ_I [9]. we develop the system (9) imposing a unique pressure on the interface as in BDD as $P_\Gamma^{(s)} = P_\Gamma$ to obtain :

$$M_{uu}^{(s)} U_\Gamma^{(s)} + M_{up}^{(s)} P_\Gamma + T^{(s)^T} \lambda = \tilde{F}_\Gamma^{(s)}, \quad (10)$$

$$M_{pu}^{(s)} U_\Gamma^{(s)} - M_{pp}^{(s)} P_\Gamma = -\tilde{F}^{(s)}, \quad (11)$$

$$\sum_{s=1}^{N} T^{(s)} U_\Gamma^{(s)} = 0. \tag{12}$$

We can then eliminate the degrees of freedom associated to the velocity in the equation (10) as in FETI taking into account the possibly singularity of the matrices $M_{uu}^{(s)}, s = 1, \cdots, N$. Using the previously obtained velocity into the equations (11) and (12) we get the FETI type interface system:

$$\begin{pmatrix} F_{DP} & -G_I \\ -G_I^T & 0 \end{pmatrix} \begin{pmatrix} \Lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} d \\ -e^T \end{pmatrix} \tag{13}$$

where

$$F_{DP}^{(s)} = \begin{pmatrix} \left(M_{uu}^{(s)^+}\right) & \left(M_{uu}^{(s)^+}\right) M_{up}^{(s)} \\ M_{pu}^{(s)} \left(M_{uu}^{(s)^+}\right) & \left(M_{pp}^{(s)} + M_{pu}^{(s)} \left(M_{uu}^{(s)^+}\right) M_{up}^{(s)}\right) \end{pmatrix}, \quad F_{DP} = \sum_{s=1}^{N} B^{(s)} F_{DP}^{(s)} B^{(s)^T},$$

$$B^{(s)} = \begin{pmatrix} T^{(s)} & 0 \\ 0 & L^{(s)^T} \end{pmatrix}, G_I = \begin{pmatrix} T^{(1)} R_u^{(1)} & \cdots & T^{(N_f)} R_u^{(N_f)} \\ 0 & \cdots & 0 \end{pmatrix}, \quad \Lambda = \begin{pmatrix} \lambda \\ P_\Gamma \end{pmatrix},$$

N_f the number of floating subdomains, $R_u^{(s)}, s = 1, \cdots, N_f$ store the basis of the kernel of the matrices $M_{uu}^{(s)}$ and α a combination of them. The interface system (13) derives from a substructuring strategy using one-level FETI on the velocity and the primal Schur complement method on the pressure and shares some common ideas with previous methods. Indeed, the idea of combining dual and primal Schur complement method to solve algebraic systems has been introduced in [3]. A generalization of FETI and primal Schur complement has been obtained using A-FETI, a three-field variant of FETI [6]. In [8], the authors use FETI-DP on the velocity and the primal Schur complement on the pressure to solve the algebraic system arising from the discretization of Stokes with a modified Hood-Taylor element.
Interchanging the role of $U_\Gamma^{(s)}$ and $P_\Gamma^{(s)}$ we obtain the matrix

$$F_{PD}^{(s)} = \begin{pmatrix} \left(M_{uu}^{(s)} + M_{up}^{(s)} \left(M_{pp}^{(s)^+}\right) M_{pu}^{(s)}\right) & M_{up}^{(s)} \left(M_{pp}^{(s)^+}\right) \\ \left(M_{pp}^{(s)^+}\right) M_{pu}^{(s)} & \left(M_{pp}^{(s)^+}\right) \end{pmatrix}, \quad s = 1, \cdots, N. \tag{14}$$

We have

Lemma 2. *Matrices $F_{DP}^{(s)}, s = 1, \cdots, N$ are symmetric positive semi-definite.*

Proof. Matrices $F_{DP}^{(s)}, s = 1, \cdots, N$ are clearly symmetric. For any $\begin{pmatrix} \lambda^{(s)} \\ P_\Gamma^{(s)} \end{pmatrix}$ let us compute the following quantity

$$\begin{pmatrix} \lambda^{(s)} \\ P_\Gamma^{(s)} \end{pmatrix}^T F_{DP}^{(s)} \begin{pmatrix} \lambda^{(s)} \\ P_\Gamma^{(s)} \end{pmatrix} =$$

$$\begin{pmatrix}\lambda^{(s)}\\ P_\Gamma^{(s)}\end{pmatrix}^T \begin{pmatrix} T^{(s)}\left(M_{uu}^{(s)^+}\right)T^{(s)^T} & T^{(s)}\left(M_{uu}^{(s)^+}\right)M_{up}^{(s)} \\ M_{pu}^{(s)}\left(M_{uu}^{(s)^+}\right)T^{(s)^T} & \left(M_{pp}^{(s)}+M_{pu}^{(s)}\left(M_{uu}^{(s)^+}\right)M_{up}^{(s)}\right)\end{pmatrix}\begin{pmatrix}\lambda^{(s)}\\ P_\Gamma^{(s)}\end{pmatrix} =$$
$$\left\{\lambda^{(s)}+M_{up}^{(s)}P_\Gamma^{(s)}\right\}^T M_{uu}^{(s)^+}\left\{\lambda^{(s)}+M_{up}^{(s)}P_\Gamma^{(s)}\right\}+P_\Gamma^{(s)^T}M_{pp}^{(s)}P_\Gamma^{(s)}. \quad (15)$$

We have shown that matrices $M_{pp}^{(s)}$ are positive semi-definite and matrices $M_{uu}^{(s)^+}$ are known to be positive semi-definite [4]. We can then conclude by (15) that matrices $F_{DP}^{(s)}, s=1,\cdots,N$ are positive semi-definite in general.

The FETI type operator F_{DP} is thus positive semi-definite in general and we can solve the system (13) by projected preconditioned conjugate gradient [4]. The suitable projector P is a matrix that projects Λ onto the null space of G_I^T. The preconditioner we choose is BDD with a local component defined as a weighted sum of matrices $F_{PD}^{(s)}$ and a coarse problem using the possibly kernel $\begin{pmatrix}-M_{up}^{(s)}R_p^{(s)}\\ R_p^{(s)}\end{pmatrix}$ of matrices $F_{DP}^{(s)}$. Define weights $\left\{D_u^{(s)}\right\}_{s=1,N}$ and $\left\{D_p^{(s)}\right\}_{s=1,N}$ associated with velocity and pressure respectively and the matrices

$$C=\begin{pmatrix}-D_u^{(1)}M_{up}^{(1)}R_p^{(1)}\cdots-D_u^{(N)}M_{up}^{(N)}R_p^{(N)}\\ D_p^{(1)}R_p^{(1)}\cdots D_p^{(N)}R_p^{(N)}\end{pmatrix},$$
$$B_D^{(s)}=\begin{pmatrix}D_u^{(s)}T^{(s)} & 0\\ 0 & L^{(s)^T}D_p^{(s)}\end{pmatrix},\quad s=1,\cdots,N.$$

The BDD algorithm is defined as follows:

3 Theoretical analysis of the condition number

Define $T=\sum_{s=1}^N B_D^{(s)}F_{PD}^{(s)}B_D^{(s)^T}$ and P_0 the $P^TF_{DP}P-$ orthogonal projection on the kernel of $F_{DP}^{(s)}$. Following [9] one can prove

Lemma 3. *The algorithm above returns* $z=M\begin{pmatrix}r_u\\ r_p\end{pmatrix}$, *where*

$$M=\left((Id-P_0)\,T\left(P^TF_{DP}P\right)(Id-P_0)+P_0\right)\left(P^TF_{DP}P\right)^{-1}. \quad (20)$$

We have

Theorem 1. *The algorithm above returns* $z=M\begin{pmatrix}r_u\\ r_p\end{pmatrix}$, *where* M *is a symmetric positive definite matrix and cond* $\left(M,P^TF_{DP}P\right)\leq c$, *where*

(i) Balance the original residual $\begin{pmatrix} r_u \\ r_p \end{pmatrix}$ by solving the auxiliary problem

$$C^T P^T F_{DP} P C \mu = C^T \begin{pmatrix} r_u \\ r_p \end{pmatrix}, \tag{16}$$

(ii) Compute the matrix-vector product

$$\begin{pmatrix} \bar{\lambda}^{(s)} \\ \bar{P}_\Gamma^{(s)} \end{pmatrix} = F_{PD}^{(s)} B_D^{(s)^T} \left(\begin{pmatrix} r_u \\ r_p \end{pmatrix} - P^T F_{DP} C \mu \right), \quad s = 1, \cdots, N, \tag{17}$$

(iii) Balance the residual by solving the coarse problem

$$C^T F_{DP} C \gamma = C^T \left(\begin{pmatrix} r_u \\ r_p \end{pmatrix} - P^T F_{DP} P \sum_{s=1}^{N} B_D^{(s)} \begin{pmatrix} \bar{\lambda}^{(s)} \\ \bar{P}_\Gamma^{(s)} \end{pmatrix} \right), \tag{18}$$

(iv) Average the solutions on the interface

$$M \begin{pmatrix} r_u \\ r_p \end{pmatrix} = \sum_{s=1}^{N} B_D^{(s)} \begin{pmatrix} \bar{\lambda}^{(s)} \\ \bar{P}_\Gamma^{(s)} \end{pmatrix} + C\gamma. \tag{19}$$

$$c = \sup \left\{ \frac{\sum_{s=1}^{N} \left\| B^{(s)^T} P \sum_{r=1}^{N} B_D^{(r)} \begin{pmatrix} \hat{\lambda}^{(r)} \\ \hat{P}_\Gamma^{(r)} \end{pmatrix} \right\|_{F_{DP}^{(s)}}^2}{\sum_{s=1}^{N} \left\| \begin{pmatrix} \hat{\lambda}^{(s)} \\ \hat{P}_\Gamma^{(s)} \end{pmatrix} \right\|_{F_{DP}^{(s)}}^2} : G_I^T \begin{pmatrix} \hat{\lambda}^{(s)} \\ \hat{P}_\Gamma^{(s)} \end{pmatrix} = 0, \right.$$
$$\left. \left\langle \begin{pmatrix} \hat{\lambda}^{(s)} \\ \hat{P}_\Gamma^{(s)} \end{pmatrix}, \begin{pmatrix} \hat{\mu}^{(s)} \\ \hat{Q}_\Gamma^{(s)} \end{pmatrix} \right\rangle = 0, \forall \begin{pmatrix} \hat{\mu}^{(s)} \\ \hat{Q}_\Gamma^{(s)} \end{pmatrix} \in Ker(F_{DP}^{(s)}), \quad 1 \le s \le N \right\}. \tag{21}$$

We omit the proof of the theorem above because it essentially follows [9].

4 Conclusion

We have combined FETI and BDD to solve the discrete Stokes with continuous pressure. The original system is reduced to an interface system whose matrix is symmetric positive semi-definite in general and whose unknowns are physically homogeneous. We have given the operator form of the preconditioner and a result from which a bound for the condition number could be derived.

References

1. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005). DOI 10.1017/S0962492904000212
2. Calgaro, C., Laminie, J.: On the domain decomposition method for the generalized Stokes problem with continuous pressure. Numer. Methods Partial Differential Equations **16**(1), 84–106 (2000). DOI 10.1002/(SICI)1098-2426(200001)16:1⟨84::AID-NUM7⟩3.0.CO;2-2
3. Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., Rixen, D.: FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. Internat. J. Numer. Methods Engrg. **50**(7), 1523–1544 (2001). DOI 10.1002/nme.76
4. Farhat, C., Roux, F.X.: Implicit parallel processing in structural mechanics. Comput. Mech. Adv. **2**(1), 124 (1994)
5. Girault, V., Raviart, P.A.: Finite element methods for Navier-Stokes equations, *Springer Series in Computational Mathematics*, vol. 5. Springer-Verlag, Berlin (1986). DOI 10.1007/978-3-642-61623-5. Theory and algorithms
6. Gosselet, P., Rey, C.: Non-overlapping domain decomposition methods in structural mechanics. Arch. Comput. Methods Engrg. **13**(4), 515–572 (2006). DOI 10.1007/BF02905857
7. Li, J.: A dual-primal FETI method for incompressible Stokes equations. Numer. Math. **102**(2), 257–275 (2005). DOI 10.1007/s00211-005-0653-y
8. Li, J., Tu, X.: A Nonoverlapping Domain Decomposition Method for Incompressible Stokes Equations with Continuous Pressures. SIAM J. Numer. Anal. **51**(2), 1235–1253 (2013). DOI 10.1137/120861503
9. Mandel, J.: Balancing domain decomposition. Comm. Numer. Methods Engrg. **9**(3), 233–241 (1993). DOI 10.1002/cnm.1640090307

Stable computations of generalized inverses of positive semidefinite matrices

A. Markopoulos[1], Z. Dostál[1], T. Kozubek[1], P. Kovář[1], T. Brzobohatý[1], and R. Kučera[1]

1 Introduction

Due to the rounding errors, effective elimination of the displacements of "floating" subdomains is a nontrivial ingredient of implementation of FETI methods, as it can be difficult to recognize the positions of zero pivots when the nonsingular diagonal block of $\mathbf{A}$ is ill-conditioned. Moreover, even if the zero pivots are recognized properly, it turns out that the ill-conditioning of the nonsingular submatrix defined by the nonzero pivots can have a devastating effect on the precision of the solution.

Most of the results are related to the first problem, i.e., to identify reliably the zero pivots. Thus [6] proposed to combine the Cholesky decomposition with the singular value decomposition (SVD) of the related Schur complement $\mathbf{S}$ in order to guarantee a proper rank of the generalized inverse. A natural modification of their method is to carry out the Cholesky decomposition as long as sufficiently large pivots are generated, and then to switch to SVD of $\mathbf{S}$. The dimension of $\mathbf{S}$ is typically small, not greater than four for 2D problems or $3m+3$ for 3D problems of linear elasticity, where m is the number of the last nodes that can be placed on a line.

Here we review our results [2, 5] related to the solution of SPS systems arising in FETI methods. In particular in the Total FETI, a variant [4] of the FETI domain decomposition method that implements both prescribed displacements and interface conditions by the Lagrange multipliers, so that the kernels of the stiffness matrices of the subdomains, i.e., their rigid body motions, are known a priori. We show, using a suitable (left) generalized inverse, how to reduce the solution of local SPS systems to the decomposition of an a priori defined well-conditioned positive definite diagonal block $\mathbf{A}_{JJ}$ of $\mathbf{A}$ and application of a suitable generalized inverse of its Schur complement $\mathbf{S}$. Since the Schur complement $\mathbf{S}$ in our approach is typically very small, the generalized inverse can be effectively evaluated by the SVD. If the rank of $\mathbf{A}$ or a lower bound on the nonzero eigenvalues of $\mathbf{A}$ are known, as happens in the implementation of TFETI, then the SVD can be implemented without any "epsilon". Moreover, if the kernel of $\mathbf{A}$ is known, then the SVD decomposition can be replaced by effective regularization. Alternatively, we show ([5]) that the kernel can be used to identify a reasonably conditioned nonsingular submatrix of $\mathbf{A}$ of the maximal order, so that $\mathbf{S} = \mathbf{O}$. Our method can be considered as a variant

[1] Centre of Excellence IT4I, VŠB-Technical University of Ostrava, Czech Republic, 17. listopadu, 15/2172, 708 33 Ostrava - Poruba, e-mail: {alexandros.markopoulos}{zdenek.dostal}{tomas.kozubek}{petr.kovar}{tomas.brzobohaty}{radek.kucera}@vsb.cz

of the regularization method or the LU–SVD method of [6] with a priori choice of the well-conditioned nonsingular part of **A** based on a combination of mechanical and combinatorial arguments. Related methods which use an information from the kernel to determine the positions of zero pivots were also proposed by [10, 1].

We review also results of [8], where we proposed a regularization technique enabling us to define a non-singular matrix $\mathbf{A}_\rho$ whose inverse is the generalized inverse to **A**. It avoids the necessity to identify zero pivots. The favorable feature of our regularization is that an extra fill-in effect in the pattern of the matrix may be negligible.

2 Cholesky decomposition and fixing nodes

We assume that **A** is an SPS stiffness matrix of a "floating" 2D or 3D elastic body, such as a subdomain in the TFETI method. If we choose M of the total N mesh nodes that are neither near each other nor placed near any line, $M < N$, $M \geq 2$ in 2D, and $M \geq 3$ in 3D, then the submatrix $\mathbf{A}_{JJ}$ of the stiffness matrix **A** defined by the set J with the indices of the displacements of the other nodes is "reasonably" nonsingular. This is not surprising, as $\mathbf{A}_{JJ}$ can be considered as the stiffness matrix of the body that is fixed in the chosen nodes. It is natural to assume that if fixing of the chosen nodes makes the body stiff, then $\mathbf{A}_{JJ}$ is well-conditioned. We call the M chosen nodes *fixing nodes* and denote by I the set of indices of corresponding displacements. In this section, we show how to combine this observation with the regularization of the Schur complement ([11]) or with the LU–SVD method proposed by [6].

Our starting point is the following decomposition of the SPS matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$

$$\widetilde{\mathbf{A}} = \mathbf{P}\mathbf{A}\mathbf{P}^T = \begin{bmatrix} \widetilde{\mathbf{A}}_{JJ} & \widetilde{\mathbf{A}}_{JI} \\ \widetilde{\mathbf{A}}_{IJ} & \widetilde{\mathbf{A}}_{II} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{JJ} & \mathbf{O} \\ \mathbf{L}_{IJ} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{JJ}^T & \mathbf{L}_{IJ}^T \\ \mathbf{O} & \mathbf{S} \end{bmatrix}, \tag{1}$$

where $\mathbf{L}_{JJ} \in \mathbb{R}^{r\times r}$ is a lower factor of the Cholesky decomposition of $\widetilde{\mathbf{A}}_{JJ}$, $\mathbf{L}_{IJ} \in \mathbb{R}^{s\times r}$, $r = n - s$, $s = 2M$ in 2D, $s = 3M$ in 3D, $\mathbf{L}_{IJ} = \widetilde{\mathbf{A}}_{IJ}\mathbf{L}_{JJ}^{-T}$, **P** is a permutation matrix, and $\mathbf{S} \in \mathbb{R}^{s\times s}$ is the Schur complement matrix defined by

$$\mathbf{S} = \widetilde{\mathbf{A}}_{II} - \widetilde{\mathbf{A}}_{IJ}\widetilde{\mathbf{A}}_{JJ}^{-1}\widetilde{\mathbf{A}}_{JI}.$$

To find **P**, we proceed in two steps. First we form a permutation matrix $\mathbf{P}_1$ to decompose **A** into blocks

$$\mathbf{P}_1\mathbf{A}\mathbf{P}_1^T = \begin{bmatrix} \mathbf{A}_{JJ} & \mathbf{A}_{JI} \\ \mathbf{A}_{IJ} & \mathbf{A}_{II} \end{bmatrix}, \tag{2}$$

where the submatrix $\mathbf{A}_{JJ}$ is nonsingular and $\mathbf{A}_{II}$ corresponds to the degrees of freedom of the M fixing nodes. Then we apply a suitable reordering algorithm on $\mathbf{P}_1\mathbf{A}\mathbf{P}_1^T$

to get a permutation matrix $\mathbf{P}_2$ which leaves the part $\mathbf{A}_{II}$ without changes and enables the sparse Cholesky decomposition of $\mathbf{A}_{JJ}$. Further, we decompose $\mathbf{PAP}^T$ as shown in (1) with $\mathbf{P} = \mathbf{P}_2\mathbf{P}_1$. To preserve sparsity we use any sparse reordering algorithm such as symmetric approximate minimum degree, symmetric reverse Cuthill-McKee, profile and wavefront reduction etc. The choice depends on the way in which the sparse matrix is stored and on the problem geometry. It is easy to verify that

$$\mathbf{A}^+ = \mathbf{P}^T \begin{bmatrix} \mathbf{L}_{JJ}^{-T} & -\mathbf{L}_{JJ}^{-T}\mathbf{L}_{IJ}^T\mathbf{S}^+ \\ \mathbf{O} & \mathbf{S}^+ \end{bmatrix} \begin{bmatrix} \mathbf{L}_{JJ}^{-1} & \mathbf{O} \\ -\mathbf{L}_{IJ}\mathbf{L}_{JJ}^{-1} & \mathbf{I} \end{bmatrix} \mathbf{P}, \tag{3}$$

where $\mathbf{S}^+ \in \mathbb{R}^{s\times s}$ denotes a left generalized inverse which satisfies

$$\mathbf{S} = \mathbf{S}\mathbf{S}^+\mathbf{S}.$$

Since s is small, we can substitute for $\mathbf{S}^+$ the Moore–Penrose generalized inverse $\mathbf{S}^\dagger \in \mathbb{R}^{s\times s}$ computed by the SVD. To see that $\mathbf{S}^\dagger$ can be evaluated effectively, first observe that the eigenvectors of $\mathbf{S}$ that correspond to the zero eigenvalues are the traces of the vectors from the kernel of $\mathbf{A}$ on the fixing nodes. Indeed, if $\widetilde{\mathbf{A}}\mathbf{e} = \mathbf{o}$, then

$$\widetilde{\mathbf{A}}_{JJ}\mathbf{e}_J + \widetilde{\mathbf{A}}_{JI}\mathbf{e}_I = \mathbf{o}, \qquad \widetilde{\mathbf{A}}_{IJ}\mathbf{e}_J + \widetilde{\mathbf{A}}_{II}\mathbf{e}_I = \mathbf{o},$$

and

$$\mathbf{S}\mathbf{e}_I = (\widetilde{\mathbf{A}}_{II} - \widetilde{\mathbf{A}}_{IJ}\widetilde{\mathbf{A}}_{JJ}^{-1}\widetilde{\mathbf{A}}_{JI})\mathbf{e}_I = \widetilde{\mathbf{A}}_{II}\mathbf{e}_I - \widetilde{\mathbf{A}}_{IJ}\widetilde{\mathbf{A}}_{JJ}^{-1}(-\widetilde{\mathbf{A}}_{JJ}\mathbf{e}_J) = \mathbf{o}. \tag{4}$$

Thus if we know the defect d of $\mathbf{A}$, which is the case in the problems arising from application of the TFETI method, we can replace d smallest nonzero eigenvalues of $\mathbf{S}$ by zeros to get the best approximation of $\mathbf{S}$ with the correct rank $s-d$. Alternatively, we can identify the zero eigenvalues correctly if we know a lower bound c on the smallest nonzero eigenvalues of $\mathbf{A}$. Due to the Schur complement eigenvalue interlacing property proved by [12], it follows that the nonzero eigenvalues of $\mathbf{S}$ are also greater or equal to c, so we can replace the computed eigenvalues of $\mathbf{S}$ that do not exceed c by zeros to get an approximation of $\mathbf{S}$ that complies with our information on $\mathbf{A}$. If neither is the case, it seems that the best we can do is to choose some small ε and to replace the eigenvalues that are smaller than ε by zeros (see, e.g., [6, 10]).

It follows from (4) that the kernel of $\mathbf{S}$ is spanned by the trace of a basis of the kernel of $\mathbf{A}$ on the fixing nodes. Assume that the kernel of $\mathbf{A}$ is known, i.e., we know $\mathbf{R} \in \mathbb{R}^{n\times d}$ whose columns span the kernel of $\mathbf{A}$. Assembling $\mathbf{R}_{I*}$ by Ith rows of $\mathbf{R}$, we define the orthogonal projector onto the kernel of $\mathbf{S}$ by

$$\mathbf{Q} = \mathbf{R}_{I*}\left(\mathbf{R}_{I*}^T\mathbf{R}_{I*}\right)^{-1}\mathbf{R}_{I*}^T$$

and we replace $\mathbf{S}^+$ in (3) by

$$\mathbf{S}^* = (\mathbf{S} + \rho\mathbf{Q})^{-1} = \mathbf{S}^\dagger + \rho^{-1}\mathbf{Q}, \quad \rho > 0.$$

We use $\rho \approx \|\mathbf{A}\|$. To see that $\mathbf{S}^*$ is a left generalized inverse, notice that

$$\mathbf{S}\mathbf{S}^*\mathbf{S} = \mathbf{S}\left(\mathbf{S}+\rho\mathbf{Q}\right)^{-1}\mathbf{S} = \mathbf{S}\left(\mathbf{S}^\dagger+\rho^{-1}\mathbf{Q}\right)\mathbf{S} = \mathbf{S}\mathbf{S}^\dagger\mathbf{S}+\rho^{-1}\mathbf{S}\mathbf{Q}\mathbf{S} = \mathbf{S}.$$

Such approach can be considered as a variant of regularization by [11]. In the next section, we show how to carry out the regularization directly on $\mathbf{A}$.

3 Regularization

This section deals with generalized inverses, for which the necessity to recognize zero pivots is avoided. We regularize $\mathbf{A} \in \mathbb{R}^{n\times n}$ using the known matrix $\mathbf{R} \in \mathbb{R}^{n\times d}$ whose columns span the kernel of $\mathbf{A}$. Although our regularization is general, i.e., it works for rectangular matrices (see [8]), we confine ourself to the SPS matrix $\mathbf{A}$.

Let us introduce the matrix $\mathbf{M} \in \mathbb{R}^{n\times d}$ so that $\mathbf{M}^\top\mathbf{R}$ is nonsingular. Let us assemble to $\mathbf{A}$ the regularized matrix $\mathbf{A}_\rho$ as follows:

$$\mathbf{A}_\rho = \mathbf{A}+\rho\mathbf{M}\mathbf{M}^\top, \tag{5}$$

where $\rho > 0$ is fixed. The following results are proved in [8].

Theorem 1. *The matrix $\mathbf{A}_\rho$ is symmetric, positive definite (and non-singular) and its inverse $\mathbf{A}_\rho^{-1}$ is the generalized inverse to $\mathbf{A}$.*

Remark 1. If $\mathbf{M} = \mathbf{R}$, we can get the Moore-Penrose inverse $\mathbf{A}^\dagger$ to $\mathbf{A}$ by

$$\mathbf{A}^\dagger = \mathbf{A}_\rho^{-1}\mathbf{P}_{ImA}, \tag{6}$$

where $\mathbf{P}_{ImA} = \mathbf{I}-\mathbf{R}(\mathbf{R}^\top\mathbf{R})^{-1}\mathbf{R}$ is the orthogonal projector on the image of $\mathbf{A}$.

Remark 2. If $\mathbf{A}^+$ is an arbitrary generalized inverse to $\mathbf{A}$, then the Moore-Penrose inverse $\mathbf{A}^\dagger$ is given by

$$\mathbf{A}^\dagger = \mathbf{P}_{ImA}\mathbf{A}^+\mathbf{P}_{ImA} \tag{7}$$

where $\mathbf{P}_{ImA}$ is the same as in Remark 1.

Using (7), one can prove that FETI type algorithms are invariant to the choice of generalized inverses in the sense that each generalized inverse is internally adapted to the Moore-Penrose one [8]. On the other hand, the Moore-Penrose inverse may be directly used in computations via the formulas (6) and (7). Although it should not affect the behavior of the FETI algorithm, it may stabilize computations for numerically unstable problems; see [9] for the experimental example.

Let us return to computational aspects of the regularization (5). To construct the regularization term, we use again fixing nodes, in which we fix only some DOFs to keep the sparsity pattern of $\mathbf{A}$ in $\mathbf{A}_\rho$ as small as possible (see Fig. 1). Let us denote

the set of indices of the fixing DOFs by I and the set of remaining indices by J. We assemble $\mathbf{M}$ as follows:

$$\mathbf{M} = \widetilde{\mathbf{M}}\mathbf{T}, \quad \widetilde{\mathbf{M}}_{i,:} = \begin{cases} \mathbf{R}_{i,:}, & i \in I, \\ 0, & i \in J, \end{cases}, i = 1, \ldots, k, \tag{8}$$

where $\mathbf{R}_{i,:}$ denotes the ith row of $\mathbf{R}$ and $\mathbf{T}$ is a nonsingular matrix which orthonormalizes columns of $\widetilde{\mathbf{M}}$ to protect the condition number of $\mathbf{A}_\rho$. Obviously, $\mathbf{T}$ can be efficiently computed as the upper triangular factor of the Cholesky decomposition of $\widetilde{\mathbf{M}}^\top \widetilde{\mathbf{M}}$. Finally, ρ is chosen as the maximum diagonal entry of $\mathbf{A}$ that lays between the minimum and maximum nonzero eigenvalues of $\mathbf{A}$.

The factorization $\mathbf{A}_\rho = \mathbf{L}\mathbf{L}^\top$ can be computed by the Cholesky algorithm for nonsingular matrices. The inverse $\mathbf{A}_\rho^{-1}$ (and the generalized inverse) is given by $\mathbf{A}_\rho^{-1} = \mathbf{L}^{-\top}\mathbf{L}^{-1}$. The computational complexity for band matrices is analyzed in [8]. For the sparse matrices we use a sparse Cholesky factorization in the form $\mathbf{A}_\rho = \mathbf{P}\mathbf{L}\mathbf{L}^\top\mathbf{P}^\top$, where $\mathbf{P}$ is the permutation matrix minimizing fill-in using a suitable reordering algorithm. The action of $\mathbf{A}_\rho^{-1}$ on a vector $\mathbf{v}$ is implemented as follows: $\mathbf{A}_\rho^{-1}\mathbf{v} = \mathbf{P}(\mathbf{L}^{-\top}(\mathbf{L}^{-1}(\mathbf{P}^\top\mathbf{v})))$, where the actions of $\mathbf{L}^{-\top}$ and $\mathbf{L}^{-1}$ are evaluated efficiently using backward and forward substitutions, respectively.

4 Choice of fixing nodes

To get M uniformly distributed fixing nodes we combine a mesh partitioning algorithm with a method for finding mesh centers. The algorithm reads as follows.

ALGORITHM ([2]) Given a mesh and $M > 0$.

(i) Split the mesh into M submeshes using the mesh partitioning algorithm.
(ii) Verify whether the resulting submeshes are connected. If not, a graph post-processing may be used to get connected submeshes.
(iii) Take a node lying near the center of each submesh.

Step 1 can be carried out by a code for graph decompositions such as METIS, while Step 3 can be efficiently performed using the so-called Perron vector (a unique nonnegative eigenvector corresponding to the largest eigenvalue of the mesh adjacency matrix) whose maximal entry enables us to approximate the center of the submesh. For more details see [2].

The number of DOFs given by M fixing nodes may be larger than the dimension of the kernel of $\mathbf{A}$. It is useful for engineering problems with complicated geometry. The usage of $\mathbf{M}$ instead of $\mathbf{R}$ in the regularization technique of Section 3 enables us to analyse cases when the most rows of $\mathbf{R}$ are replaced by zeros in $\mathbf{M}$. Then the regularization term in $\mathbf{A}_\rho$ influeces only few entries of $\mathbf{A}$.

5 Cholesky decomposition and the kernel of A

If the kernel of $\mathbf{A}$ is known, then we can use it to identify a submatrix $\mathbf{A}_{JJ}$ of $\mathbf{A}$ of a maximal order. Since the Schur complement of $\mathbf{A}_{JJ}$ is the zero matrix, the solution of a consistent system with $\mathbf{A}$ reduces to the Cholesky decomposition of $\mathbf{A}_{JJ}$. The following estimate proved in [5] indicates that we can use information obtained from the kernel of $\mathbf{A}$ to identify suitable zero pivots.

Proposition 1. *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ denote a symmetric matrix whose kernel is spanned by the full column rank matrix $\mathbf{R} \in \mathbb{R}^{n\times d}$ with orthonormal columns, so that d is the defect of $\mathbf{A}$. Let $I = \{i_1, \ldots, i_d\}, \quad 1 \le i_1 < i_2 < \cdots < i_d \le n$, denote a set of indices, and let $J = \mathscr{N} - I, \quad \mathscr{N} = \{1, 2, \ldots, n\}$. Then*

$$\lambda_{\min}(\mathbf{A}_{JJ}) \ge \overline{\lambda}_{\min}(\mathbf{A})\, \sigma^4_{\min}(\mathbf{R}_{I*}), \tag{9}$$

where $\overline{\lambda}_{\min}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{R}_{I*})$ denote the least nonzero eigenvalue of $\mathbf{A}$ and the least singular value of $\mathbf{R}_{I*}$.

This strategy chooses d fixing DOFs by the orthonormalization of $\mathbf{R}$ and applying the Gaussian elimination with complete pivoting to transform orthonormalized matrix $\mathbf{R}$ into the column-wise echelon form. The position of the first nonzero entry in each column gives the degree of freedom which will be fixed. For more details we refer to [5].

6 Numerical examples

The performance of our strategies is tested on the stiffness matrix $\mathbf{A}$ of the elastic three-dimensional cube made of steel and discretized by trilinear bricks with the Neumann boundary conditions (see Fig. 1.(a)). To illustrate the effect of fix-

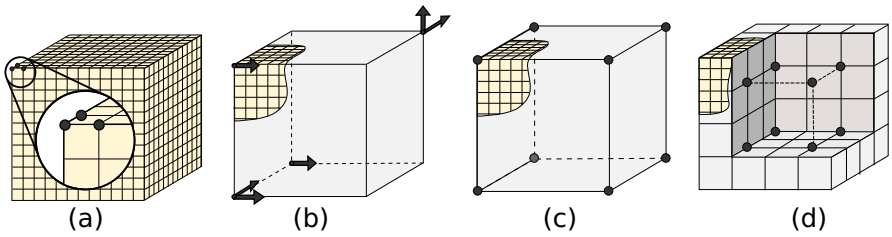


Fig. 1 (a) No strategy, (b) GP strategy, (c) Geometrical strategy, (d) Uniform strategy

ing nodes, we carried out the computations for different strategies of choosing fixing nodes depicted in Fig. 1. Here *Geometrical strategy* is the simplest one and

is based on finding fixing nodes using simple geometrical and combinatorial arguments: choose M mesh nodes that are mutually as far apart as possible and that are not placed near any line.

In Table 1, we report the regular condition number $\overline{\mathrm{cond}}(\mathbf{A})$ (ratio of the largest and the smallest nonzero eigenvalues), the condition number of the nonsingular part $\mathbf{A}_{JJ}$ decomposed by the Cholesky decomposition, and the regular condition number $\overline{\mathrm{cond}}(\mathbf{A}^+)$. The results of experiments agree with the intuitive rule that fixing nodes distributed in a more regular pattern improves the conditioning of $\mathbf{A}_{JJ}$. In particular, comparing variants (c) and (d), we can observe that placing the eight fixing nodes inside the body can result in more stable generalized inverse than placing them at the corners. It follows that the matrices arising in the original FETI method or its TFETI variant are typically better conditioned than those arising in the FETI–DP. Notice that the worst conditioning of $\mathbf{A}_{JJ}$ and $\mathbf{A}^+$ can be observed in variant (a) which is a possible result of the default strategy used by Farhat and Géradin [6].

Table 1 Characteristics of $\mathbf{A}$ and $\mathbf{A}^+$ in dependence on the distribution of fixing nodes.

	No strategy	GP strategy	Geometrical strategy	Uniform strategy
$\overline{\mathrm{cond}}(\mathbf{A})$	4.91E+02	4.91E+02	4.91E+02	4.91E+02
$\mathrm{cond}(\mathbf{A}_{JJ})$	2.90E+07	3.52E+05	9.92E+03	1.90E+03
$\overline{\mathrm{cond}}(\mathbf{A}^+)$	2.55E+07	3.52E+05	1.32E+04	1.90E+03

Table 2 shows results of numerical tests based on the regularization. The rows *iter* or *iter*† report iterations of the TFETI algorithm for the regularizations computed by strategies (b)-(d) or by the Moore-Penrose inverse obtained from them using (7), respectively. It confirms invariancy with respect to the choice of the generalized inverse. The condition numbers in the next two rows agree with the same heuristic as in Table 1, i.e., the conditioning of A_ρ^{-1} is improved when the fixing DOFs are distributed in a more regular pattern. The CPU times in the fifth and sixth rows required for computing the Cholesky decomposition and the actions of the generalized inverses, respectively, illustrate the computational invariancy that is due to the negligible fill-in. It is seen from the number of non-zero entries in the last row of the table.

Acknowledgements This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the project SPOMECH - Creating a multidisciplinary R&D team for reliable solution of mechanical problems, reg. no. CZ.1.07/2.3.00/20.0070 within Operational Programme 'Education for competitiveness' funded by Structural Funds of the European Union and state budget of the Czech Republic.

Table 2 Characteristics of $\mathbf{A}$ and $\mathbf{A}_\rho^{-1}$ in dependence on the distribution of fixing nodes.

	GP strategy	Geometrical strategy	Uniform strategy
iter	22	22	22
iter†	22	22	22
$\overline{\text{cond}}(\mathbf{A})$	4.91E+02	4.91E+02	4.91E+02
$\text{cond}(\mathbf{A}_\rho^{-1})$	3.53e+05	1.30e+04	3.02e+03
chol [sec.]	0.2897	0.2750	0.2567
action [sec.]	0.0215	0.0209	0.0210
nnz_chol	2775956	2762089	2690104

References

1. Peter Arbenz and Zlatko Drmač. On positive semidefinite matrices with known null space. *SIAM J. Matrix Anal. Appl.*, 24(1):132–149 (electronic), 2002.
2. T. Brzobohatý, Z. Dostál, T. Kozubek, P. Kovář, A. Markopoulos. Cholesky decomposition with fixing nodes to stable computation of a generalized inverse of the stiffness matrix of a floating structure. *Internat. J. Numer. Methods Engrg.*, 88(5):493-509 (electronic), 2011.
3. Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 2005.
4. Z. Dostál, T. Kozubek, A. Markopoulos, T. Brzobohatý, V. Vondrák, and P. Horyl. A theoretically supported scalable TFETI algorithm for the solution of multibody 3d contact problems with friction. *Comput Method Appl M*, 205-208:110–120, 2012.
5. Zdeněk Dostál, Tomáš Kozubek, Alexandros Markopoulos, and Martin Menšík. Cholesky decomposition of a positive semidefinite matrix with known kernel *Appl. Math. Comput.*, 217 (13):6067–6077, 2011.
6. Charbel Farhat and Michel Géradin. On the general solution by a direct method of a large-scale singular system of linear equations: application to the analysis of floating structures. *Internat. J. Numer. Methods Engrg.*, 41(4):675–696, 1998.
7. T. Kozubek, A. Markopoulos, T. Brzobohatý, R. Kučera, V. Vondrák, and Z. Dostál. Matsol - matlab efficient solvers for problems in engineering.
8. R. Kučera, T. Kozubek, A. Markopoulos. On large-scale generalized inverses in solving two-by-two block linear systems. *Linear Algebra Appl.*, 438:3011–3029, 2013.
9. R. Kučera, T. Kozubek, A. Markopoulos, J. Machalová. On the Moore-Penrose inverse in solving saddle-point systems with singular diagonal blocks. *Numerical Linear Algebra with Applications*, 19:677–699, 2012.
10. M Papadrakakis and Y Fragakis. An integrated geometric-algebraic method for solving semi-definite problems in structural mechanics. *Comput. Meth. Appl. Mech. Eng.*, 190(49–50):6513–6532, 2001.
11. E Savenkov, H Andrä, and O Iliev. An analysis of one regularization approach for solution of pure Neumann problem. Technical Report 137, Berichte des Faruenhofer ITWM, Kaiserslautern, 2008.
12. Ronald L. Smith. Some interlacing properties of the Schur complement of a Hermitian matrix. *Linear Algebra Appl.*, 177:137–144, 1992.

Parallel implementation of Total-FETI DDM with application to medical image registration

Michal Merta[1], Alena Vašatová[1], Václav Hapla[1], and David Horák[1]

1 Introduction

The main task of image registration is to determine an optimal spatial transformation such that two (or more) images become, in a certain sense, similar. Therefore, it plays a crucial role in image processing if there is a need to integrate information from two (or more) source images. These images usually show the same scene, but taken at different times, from different viewpoints or by different sensors.

Image registration is used in various areas. In medical applications it serves to obtain more complete information about the patient (e.g., to monitor a progression or regression of a disease, to align pre- and post- contrast images, or to compare patient's data with anatomical atlases), to compensate a motion of a subject during medical scanning, to correct calibration differences across scanners etc. [10, 12]. For more examples of usage of medical image registration see [8].

The first attempts at medical image registration focused mainly on the processing of brain images. Hence, a rigid body approximation was sufficient, because of a relatively small possibilities for deformation inside the skull. Later, it was extended to the affine registration. However, rigid or even affine approximations are usually not sufficient for a registration of a human body. Therefore, the research in medical image processing is now focused on the development of non-rigid registration methods. One of them is the elastic registration introduced by Broit [1]. In this method, images are considered to be 2D elastic bodies. Volume forces defined from 'differences' of the two images then deform one image so that it becomes similar to the other. The disadvantage of this linear model is that it assumes small deformations. For large deformations it can be replaced by the viscous fluid model [2].

With the increasing amount of data provided by medical instruments like CT or MRI, a parallel implementation of image registration seems to be necessary. In this work we combine the method of elastic registration together with the Total-FETI method [3] to obtain scalable algorithm for registration of medical images.

[1] Centre of Excellence IT4Innovations, VŠB-Technical University of Ostrava, tř. 17. listopadu 15/2172, Ostrava, 708 33 Czech Republic, e-mail: {michal.merta}{alena.vasatova}{vaclav.hapla}{david.horak}@vsb.cz

2 Elastic registration

Image registration usually consists of three parts: choosing an appropriate transformation model, choosing a distance (similarity) measure, and optimization process. Let us use the notation from [10] and briefly describe the process.

In order to find a transformation of the template image T, such that after its application it becomes, in a certain sense, similar to the reference image R, we define a suitable distance measure $\mathscr{D}$ and minimize the distance between R and T with respect to searched transformation φ:

$$\min_{\varphi:\mathbb{R}^2\to\mathbb{R}^2} \mathscr{D}[R,T;\varphi], \tag{1}$$

where $\mathscr{D}[R,T;\varphi] := \mathscr{D}\left[R,T_\varphi\right]$.

However, this approach has its drawbacks: a solution is not necessarily unique and it actually may not exist. Thus, the problem (1) is ill-posed. Moreover, additional implicit constraints can emerge, e.g., in medical images no additional cracks or folding of the tissue are allowed (the transformation should be diffeomorphic). Both these situations can be solved by adding a regularizer [10].

Transformation model of elastic registration is based on a physical motivation that the images are two different observations of an elastic body, one before and one after a deformation. The transformation $\varphi : \mathbb{R}^2 \to \mathbb{R}^2$ is split into the identity part and the displacement $u : \mathbb{R}^2 \to \mathbb{R}^2$:

$$\varphi(x) := x - u(x). \tag{2}$$

As the regularizer we use the linearized elastic potential

$$\mathscr{P}[u] := \int_\Omega \frac{\mu}{4} \sum_{j=1}^{2}\sum_{k=1}^{2} \left(\partial_{x_j} u_k + \partial_{x_k} u_j\right)^2 + \frac{\lambda}{2}(\operatorname{div} u)^2 \,\mathrm{d}V, \tag{3}$$

where λ and μ are the Lamé parameters. The regularizer has the meaning of volume forces, which implicitly constrain the displacement to fulfill a smoothness criteria. We obtain the following regularized problem which is more suitable for a numerical realization:

$$\mathscr{J}[u] = \min_{v:\mathbb{R}^2\to\mathbb{R}^2} \mathscr{J}[v], \quad \text{where} \quad \mathscr{J}[v] := \mathscr{D}[R,T;v] + \alpha\mathscr{P}[v]. \tag{4}$$

Here, the parameter $\alpha \in \mathbb{R}^+$ controls the strength of the smoothness of the displacement versus the similiarity of the images. In the case of the elastic registration it is usually omitted, since it can be included in the Lamé parameters. Therefore, let us assume $\alpha = 1$ in what follows.

A distance measure is a cost function which determines a similarity of two images. We choose the so-called sum of squared differences (SSD):

$$\mathscr{D}[R,T;u] := \frac{1}{2}\|T_u - R\|^2_{L_2(\Omega)}, \tag{5}$$

where $T_u(x) := T(x-u(x))$. The volume forces

$$f(x,u(x)) := (R(x) - T_u(x))\nabla T_u(x), \tag{6}$$

$f : \mathbb{R}^2 \to \mathbb{R}^2$, derived from its Gâteaux derivative, push a transformed image into the direction of a reference.

Images are represented by the compactly supported mappings $R,T : \Omega \to \mathbb{R}$, where $\Omega := (0,1)^2$. $T(x)$ and $R(x)$ denote the intensities of images at the spatial position x; we set $R(x) := 0$ and $T(x) := 0$ for all $x \notin \Omega$.

By applying the Gâteaux derivative to the elastic potential (3) we obtain the Navier-Lamé operator of classical elasticity. The displacement of the elastic body and therefore the transformation of the image T is then obtained as the solution of the partial differential equation with zero Dirichlet boundary condition:

$$\begin{cases} \mu\Delta u(x) + (\lambda+\mu)\nabla \operatorname{div} u(x) = -f(x,u(x)) & \text{in } \Omega, \\ u(x) = 0 & \text{on } \partial\Omega. \end{cases} \tag{7}$$

There are several possibilities how to overcome the non-linearity of the previous equation. In the simplest case, when the difference between the reference and the template image is small enough, we set

$$f(x,u(x)) := f(x,0) = (R(x) - T(x))\nabla T(x), \tag{8}$$

and obtain a linearized problem. Otherwise, we solve the problem iteratively using the Algorithm 1. The similar algorithm is presented in [10], where the finite differ-

Algorithm 1 Fixed-point iteration for the solution of Equation (7)

$T_0(x) := T(x)$
$f_0(x) := (R(x) - T_0(x))\nabla T_0(x)$
for $k = 1$ **to** K **do**
  solve (7) for u_k with $f(x,u(x)) := f_{k-1}$
  $T_k(x) := T_{k-1}(x - u_k)$
  $f_k(x) := (R(x) - T_k(x))\nabla T_k(x)$
end for

ence method is used for the solution of the linearized problem.

We discretize the linearized problem using a finite element method with piecewise affine basis functions on triangular elements. To approximate the gradient of T_u, which is necessary for the evaluation of forces f, we use a convolution with an appropriate kernel of the Sobel operator (see, e.g., [11]). The solution can be easily parallelized by the Total-FETI method described in the following part.

3 Parallelization using Total-FETI method

The numerical solution of the linearized version of the problem (7) can be effectively parallelized by the Total-FETI (TFETI) method which is a variant of the FETI method originally proposed by Farhat et. al. [6]. The method is based on the decomposition of the spatial domain into non-overlapping subdomains. The continuity of the solution among subdomains is enforced by Lagrange multipliers. Total-FETI by Dostál et al. [3] simplifies the inversion of stiffness matrices of subdomains by using Lagrange multipliers also to enforce the Dirichlet boundary condition. Using this approach, all subdomains are floating and their stiffness matrices have the same kernels formed by the vectors of the rigid body modes.

To apply the FETI based domain decomposition, we partition the rectangular domain Ω, representing the processed image, into N geometrically identical rectangular subdomains Ω_s. We denote K_s, f_s, u_s, and B_s the subdomain stiffness matrix, the subdomain load vector, the subdomain displacement vector, and the subdomain constraint matrix, respectively. Let us also denote R_s as the matrix with columns forming the basis of the kernel of K_s. Notice, that because of this regular decomposition, the matrices K_s, as well as R_s, are the same for all subdomains. Therefore, they are computed only once and then redistributed among processors. Eventually, they can be stored in a shared memory.

After the decomposition we obtain the quadratic minimization problem with equality constraints

$$\min \frac{1}{2} u^T K u - u^T f \quad \text{s. t.} \quad Bu = c, \tag{9}$$

where

$$K := \begin{bmatrix} K_1 & & \\ & \ddots & \\ & & K_N \end{bmatrix}, \quad f := \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}, \quad u := \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix}, \quad B := [B_1, \ldots, B_N]. \tag{10}$$

Applying the duality theory to the equivalent saddle-point problem and establishing the notation

$$F := BK^{\dagger}B^T, G := R^T B^T, d := BK^{\dagger} f, e := R^T f,$$

where $K^{\dagger}$ denotes a generalised inverse matrix satisfying $KK^{\dagger}K = K$ (see, e.g., [4]), and R denotes the block-diagonal matrix with blocks R_s, we obtain the following minimization problem:

$$\min \frac{1}{2} \lambda^T F \lambda - \lambda^T d \quad \text{s.t.} \quad G\lambda = e. \tag{11}$$

We can further homogenize the equality constraints $G\lambda = e$ to $G\mu = 0$ by decomposing λ into $\mu \in \operatorname{Ker} G$ and $\tilde{\lambda} \in \operatorname{Im} G^T$ as

$$\lambda := \mu + \tilde{\lambda}. \tag{12}$$

We get $\tilde{\lambda}$ easily by $\tilde{\lambda} = G^T(GG^T)^{-1}e$. To enforce the condition $G\mu = 0$ we introduce the projector $P := I - Q$ to the null space of G. Here $Q := G^T(GG^T)^{-1}G$ is the projector onto the image space of G^T. The final problem for μ reads (note that $P\mu = \mu$):

$$PF\mu = P(d - F\tilde{\lambda}). \tag{13}$$

This problem can be effectively solved by the conjugate gradient method.

One of the advantages of the approach based on the Lagrange multipliers is the possibility to include other constraints to the matrix B than 'gluing' and Dirichlet conditions. One possibility is to use it to enforce the rigidity of certain parts of the processed image. These rigid parts can represent, e.g., bones. As mentioned in Section 2, the new coordinates $\varphi(x)$ of any point x after transformation are

$$\varphi(x) := x - u(x). \tag{14}$$

Using rigid body motions with a linearized rotation, this transformation can also be described by

$$x - u(x) = R_x a, \tag{15}$$

where

$$R_x := \begin{bmatrix} -x_2 & 1 & 0 \\ x_1 & 0 & 1 \end{bmatrix}, \tag{16}$$

and a is the vector of motion parameters (shifts and rotation). Conditions necessary to enforce a rigidity of a motion of two point $\tilde{x}, \tilde{y}$ can be derived from the following system of equations

$$\begin{cases} \tilde{x} - u(\tilde{x}) = R_{\tilde{x}} a, \\ \tilde{y} - u(\tilde{y}) = R_{\tilde{y}} a. \end{cases} \tag{17}$$

We eliminate a and obtain

$$-ou_1(\tilde{x}) - pu_2(\tilde{x}) + ou_1(\tilde{y}) + pu_2(\tilde{y}) = p^2 + o^2, \tag{18}$$

where $p := \tilde{y}_2 - \tilde{x}_2$, $o := \tilde{y}_1 - \tilde{x}_1$, and $u(x) := (u_1(x), u_2(x))$. These conditions are added to appropriate positions in the matrix B. To reduce the number of additional constraints, one can enforce the rigidity only on the boundaries of given areas.

4 Data parallelization and implementation using Trilinos framework

Parallelization of FETI/TFETI can be implemented using SPMD technique – distributing matrix portions among the processing units. The distribution of primal data is straightforward because of the block-diagonal structure of the system stiffness

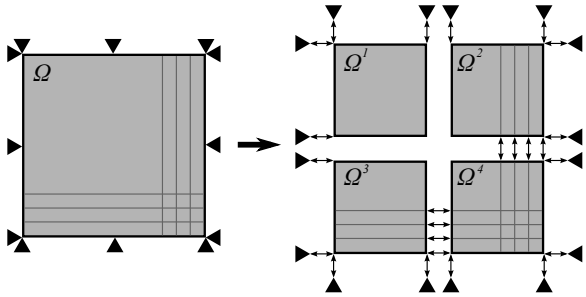


Fig. 1 Total-FETI domain decomposition of the 2D rectangular area. Dirichlet boundary conditions are enforced by Lagrange multipliers.

matrix. Each processor is assigned one rectangular part of the images R and T, and the corresponding primal data – one block of the global stiffness matrix K, one block of the kernel matrix R, and corresponding parts of the constraint matrix B, solution vector u, and right-hand side vector f. On the other hand, if we want to accelerate also the dual actions we have to distribute the dual objects as well. We distribute the matrix G into vertical blocks. All dual vectors are distributed accordingly to this (for more details see [9]).

For the parallel implementation we use the Trilinos framework [7] which is a collection of relatively independent packages developed by Sandia National Laboratories. It provides a tool kit for basic linear algebra operations (both serial and parallel), direct and iterative solvers to linear systems, PDE discretization utilities, etc. Its main advantages are object oriented design, high modularity and use of modern features of C++ language such as templating. It is currently in version 11.

In our codes we use the Epetra package as a base for linear algebra operations. It provides users with distributed dense vectors and matrices, as well as sparse matrices in compressed row format (`Epetra_CrsMatrix`), linear operators, distributed graphs, etc. As the object-oriented wrapper to direct linear system solver SuperLU, which is used for the solution of the coarse problem (application of $(GG^T)^{-1}$) and the application of the pseudoinverse $K^\dagger$, we use the Amesos package.

5 Numerical experiments

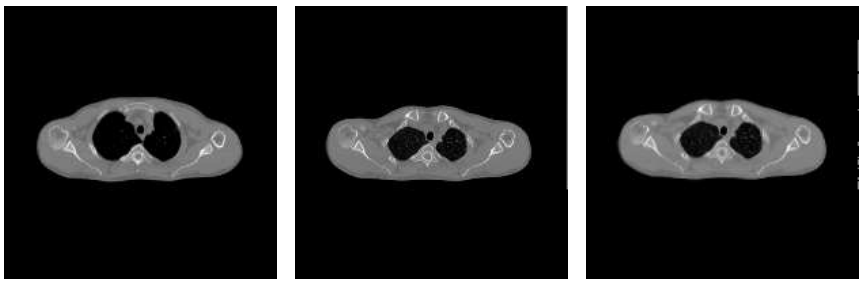
The numerical experiments were performed on the cluster consisting of 16 SMP nodes, each of the nodes is equipped with two Intel Xeon QuadCore 2.5 GHz CPUs and 18 GB of RAM. Table 1 shows the results of the scalability tests for the data obtained from Department of Oncology of University Hospital of Ostrava. We perfomed two experiments – one with no additional constraints, and the second on the same data but with a rigidity of the bones enforced by additional Lagrange multipliers. The processed data are depicted in Figure 2.

The problem is linearized using the approach (8). For the first experiment, the number of CG iterations is relatively low. For these numbers of dual variables the coarse problem (which is usually the main bottleneck of the FETI methods) is not big enough to affect the scalability and the increasing time per iteration is caused mainly by the communication and vector redistribution routines within the Trilinos framework. The second experiment shows that the additional constraints lead to the increase of the number of CG iterations. To reduce this number we can use the cheap lumped preconditioner $\overline{F^{-1}} = BKB^T$ (see [5]).

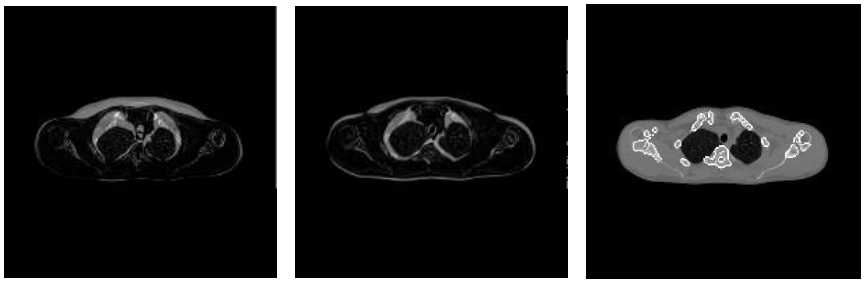
Table 1 Performance of the TFETI implementation for varying decomposition and discretization

Number of subdom.	1	4	16
Primal dimension	20,402	81,608	326,432
Dual dimension	808	2,424	8,080
CG time [s]	0.50	1.53	4.35
CG iterations	25	39	47
Time per iteration [s]	0.02	0.04	0.09
Example 1: Without rigid body parts			

Number of subdom.	1	4	16
Primal dimension	20,402	81,608	326,432
Dual dimension	903	2,641	8,254
CG time [s]	41.01	34.54	57.44
CG iterations	2467	990	665
Time per iteration [s]	0.01	0.03	0.08
Example 2: With rigid body parts			



(a) Image R (b) Image T (c) Image T_φ



(d) Difference between R and T (e) Difference between R and T_φ (f) Boundaries of rigid parts (bones)

Fig. 2 Processed data - computer tomography of patient's chest. We search for a transformation φ of the image T (in exhalation) so it becomes similar to the image R (in inflation). For this experiment, we set $\mu = 5 \times 10^5$ and $\lambda = 0$.

6 Conclusion

We have demonstrated the applicability of the Total-FETI method to a parallelization of a process of image registration. Our implementation was tested on 2D computer tomography data obtained from University Hospital of Ostrava. Because of relatively low resolution of the images the total number of unknowns in the resulting systems did not exceed hundreds of thousands. However, these results enable us to focus on the development of domain decomposition-based methods for the image registration of 3D data, where the number of unknowns can easily reach tens or hundreds of millions.

Acknowledgements This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the project SPOMECH - Creating a multidisciplinary R&D team for reliable solution of mechanical problems, reg. no. CZ.1.07/2.3.00/20.0070 within Operational Programme 'Education for competitiveness' funded by Structural Funds of the European Union and state budget of the Czech Republic.

The research has also been supported by the grants: HPC-Europa2 project funded by the European Commission - DG Research in the Seventh Framework Programme under grant agreement No. 228398, PRACE 2IP project receiving funding from the EU's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. RI-283493.

References

1. Broit, C.: Optimal registration of deformed images. Ph.D. thesis, University of Pensylvania (1998)
2. Christenssen, G.: Deformable shape models for anatomy. Ph.D. thesis, Washington University (1994)
3. Dostál, Z., Horák, D., Kučera, R.: Total feti - an easier implementable variant of the feti method for numerical solution of elliptic pde. Commun. in Numerical Methods in Engineering **22**, 1155–1162 (2006)
4. Dostál, Z., Kozubek, T., Markopoulos, A., Menšík, M.: Cholesky decomposition and a generalized inverse of the stiffness matrix of a floating structure with known null space. Applied Mathematics and Computation **217**, 6067–6077 (2011)
5. Farhat, C., Mandel, J., Roux, F.: Optimal convergence properties of the feti domain decomposition method. Comput. Methods Appl. Mech. Eng. **115**, 365–385 (1994)
6. Farhat, C., Roux, F.: An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems. SIAM Journal on Scientific Computing **13**, 379–396 (1992)
7. Heroux, M.: Trilinos web page. Available from http:://trilinos.sandia.gov (2012)
8. Hill, D., Holden, M., Hawkes, D.: Medical image registration. Physics in medicine and biology **46**, 1–45 (2001)
9. Horák, D., Hapla, V.: Tfeti coarse space projectors parallelization strategies. Lecture Notes in Computer Science. **7203**, 152–162 (2012)
10. Modersitzki, J.: Numerical Methods for Image Registration. Oxford University Press, Oxford (2004)
11. Pratt, W.K.: Digital Image Processing: PIKS Inside. John Wiley & Sons, Inc., New York (2001)
12. Zitova, B., Flusser, J.: Image registration methods: a survey. Image and Vision Computing **21**, 977–1000 (2003)

Finite Element Analysis of Multi - Component Assemblies: CAD - based Domain Decomposition

Kirill Pichon Gostaf[1], Olivier Pironneau[1], and François-Xavier Roux[1]

Abstract: We apply domain decomposition to carry out finite element simulations of multi-component computer aided design (CAD) assemblies. The novelty of our research is the CAD-based domain decomposition. We consider design parts as independent sub-domains and reuse assembly topology to define regions, where the interface boundary conditions should be applied. The Dirichlet-Neumann [1], Neumann-Neumann [2] and FETI [3] methods for non-matching triangulations have been studied. We endorse the proposed framework with numerical experiments and we focus on the essence of its parallel implementation.

1 Introduction

Computer aided design (CAD) and finite element (FE) modeling are standards in a concept to manufacture industrial chain. Realistic FE simulations require huge computational resources and may last unacceptably long. In this paper, we present a comprehensive framework that allows to automate and parallelize numerical simulations of multi-component CAD assemblies. We refer to the work of Pironneau [4] et al., where the authors have proposed to use constructive solid geometry modeling as a basis for spatial domain decomposition; see [6, 7, 8, 9] for related work on three dimensional contact problems in solid mechanics.

The novelty of our research is the CAD-based domain decomposition method. We consider design parts as independent sub-domains. Then we reuse assembly topology to define regions where the interface boundary conditions should be applied. Our motivation is to automate FE management of an existing CAD data, i.e. to update only the concerning meshes when CAD parts are modified. In addition, the method aims to regularize mathematical models when using various material properties (steel, cooper, rubber etc.). The method is inherently parallel and therefore perfectly suited for hight performance computing.

[1] Laboratoire Jacques-Louis Lions, UPMC, France, e-mail: {gostaf}{pironneau}{roux}@ljll.math.upmc.fr

2 CAD-based Domain Decomposition

Generally, a FE model of an entire CAD prototype takes several days to be properly defined. Once a pre-processing stage is completed (a global mesh, loadings and constraints are generated), adaptive refinement procedure requires communication with CAD kernel at each computational iteration. Meanwhile, engineering design changes are made on a daily basis at the CAD level, and the mesh generally may not follow the changes. Hence, the FE model cannot be updated within such timespan.

Assembly-driven decomposition

The application of assembly driven domain decomposition allows to automate the above framework. We consider each component of a CAD assembly as an independent sub-domain. Triangulations are generated independently and could be further updated. Variational formulations are then explicitly written for each sub-domain. Inter-domain continuity conditions are set according to the domain decomposition algorithm.

Let $\{P_1,...,P_s\}$ refer to a set of assembly components (design parts), with $s \geq 2$. We define $\{\Omega_1,...,\Omega_s\}$ to be a set of the corresponding computational sub-domains. An illustration is given in Fig. 1.

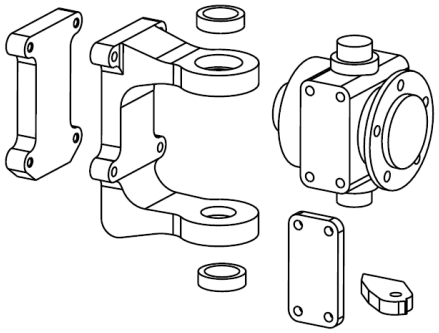


Fig. 1 An assembly-driven domain decomposition. Design parts are considered as independent computational sub-domains.

Modeling accuracy

Solid parts are generated independently of the FE process, yet they are manipulated by FE algorithms after discretization. For a manifold M (a boundary of a solid part), we define

$$H = \mathrm{diam}(M) = \sup_{x_1,x_2 \in M} |x_1 - x_2|$$

along with the "smallest feature length" l (the smallest hole, fillet, chamfer etc.). According to the CAD documentation [10], parts are initially created with the relative accuracy δ^r_{CAD} which satisfies

$$10^{-6} \leqslant l/H < \delta^r_{CAD} \leqslant 10^{-2}$$

CATIA modeling platform [11] allows to design parts with

$$10^{-6} \leqslant l, H \leqslant 10^{3}$$

however an option

$$10^{-8} \leqslant l, H \leqslant 1$$

is available to design small parts, but the module has limited implementation.

Initialization of inter-component contact regions

We propose to reuse "assembly constrains" (data on parts relative position stored in a CAD assembly file) in order to generate an initial list of the contact regions (called contact faces). Let $\mathbb{S}$ denote a set of initial contact faces between all adjacent assembly components (solid parts)

$$\mathbb{S} = \{P_i \cap^* P_j\} \qquad 1 \leq i \neq j \leq s$$

where $\cap^*$ stands for a Boolean cut operator (intersection of manifolds). The number of all possible contact pairs is bounded by the binomial coefficient

$$\dim(\mathbb{S}) \leq \binom{s}{2}$$

In practice, for CAD assemblies, the number of inter-component contact faces is much smaller than the binomial coefficient and often satisfies

$$\dim(\mathbb{S}) \sim \mathscr{O}(s)$$

Definition 1. Two objects $A \subset \mathbb{R}^d$ and $B \subset \mathbb{R}^d$, $d \geq 1$ are geometrically equal if the set A is equivalent to the set B.

Definition 2. Topological equivalence - Two objects are topologically equivalent if there is a homeomorphism between them.

In the following, a contact face $\mathscr{F}$ is a set of patches; a patch is defined by four NURBS or B-spline curves. Let $\mathscr{F}_{i,j}$ and $\mathscr{F}_{j,i}$ be the opposite contact faces belonging to the adjacent components P_i and P_j, respectively. Then, $\mathscr{T}_{i,j}$ and $\mathscr{T}_{j,i}$ be a discretization of the above contact faces. In order to build

$$\mathscr{T}_{i,j} = \mathscr{T}_{j,i} \tag{1}$$

we require both geometrical and topological equivalence of $\mathscr{F}_{i,j}$ and $\mathscr{F}_{j,i}$. However, (1) is hard to achieve, since solid models are built with only a fixed accuracy.

Remark: Obviously, matching triangulations $\mathscr{T}_{i,j} = \mathscr{T}_{j,i}$ might be generated within an additional computational cost. Unfortunately, for simulations involving sliding, mixed finite elements (shape, order) or discontinuities in material coefficients matching triangulations are hard to maintain.

Geometric discontinuities across contact faces

In practice, most contact regions are either non-planar or have curved boundaries. When meshes are generated independently, $\mathscr{T}_{i,j}$ and $\mathscr{T}_{j,i}$ often appear different, owing to round-off errors. As a result, geometric discontinuities are certain for non-matching triangulations, which is clearly seen in Fig. 2.

Let u_{1h} and u_{2h} be discrete functions defined on the triangulations $\mathscr{T}_{1,2}$ and $\mathscr{T}_{2,1}$, respectively. For $\mathscr{T}_{1,2} \neq \mathscr{T}_{2,1}$ the computation of a jump operator

$$u_{2h}(x) - u_{1h}(x)$$

on contact faces is not properly defined (not unique). Indeed, when Ω_h is a polygonal approximation of Ω, numerical integration of boundary integrals will not be equal on $\mathscr{T}_{1,2} \neq \mathscr{T}_{2,1}$. In this context, we are interested to compute the value of a finite element function $u_h(y)$ slightly outside its domain of definition, namely at $y \in \mathbb{R}^3$ close to Ω_h in the sense

$$\min_{x \in \Omega_h} |x - y| < c\,h$$

where $c \in \mathbb{R}_+$, and h is the discretization parameter (mesh size).

Assume that Ω_h is triangulated into tetrahedral elements. Let $\{v_0, ..., v_3\}$ be the vertices of a tetrahedral element T close to y. The barycentric coordinates $\{\lambda_0, ..., \lambda_3\}$ of y with respect to T satisfy

$$\sum_{i=0}^{3} \lambda_i = 1 \qquad \text{and} \qquad y = \sum_{i=0}^{3} \lambda_i\, v_i$$

When a point y does not belong to the discrete domain, we shall define

$$u_h(y) = \sum_{i=0}^{3} \lambda_i\, u_h(v_i)$$

where the vertices v_i are those of the nearest tetrahedral element. We use the same approach for a $\mathbb{P}_2$ or higher Lagrangian finite element.

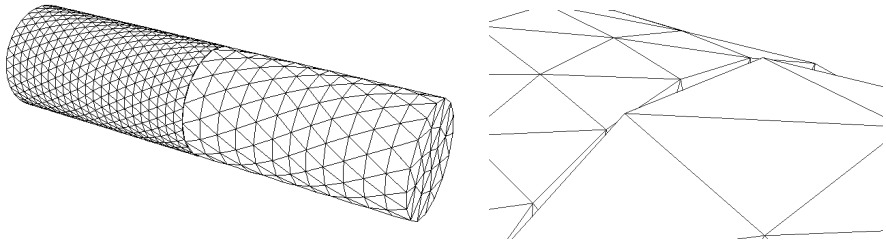


Fig. 2 Geometric discontinuity across the contact region in case of non-matching triangulations.

Parallel implementation

Depending on the computer architecture, we propose two implementation schemes: the first is suitable for small commodity clusters; the second fits massively parallel architecture (HPC). Let N_{cpu} be the number of available CPUs, processor cores, that are used for a FE simulation. Let s be the number of assembly components. Reasonably, one can expect $N_{cpu} \sim s$ for small or intermediate commodity clusters and $N_{cpu} >> s$ for HPC machines. Fig. 3 (left) illustrate the case, where three sub-domains are assigned to a single process, i.e. `cpu2`, and solved in sequence; the right chart in Fig. 3 shows the case, where each sub-domain is treated in parallel. Inside one sub-domain either algebraic mesh partitioning or multi-threading is used to parallelize a local solver.

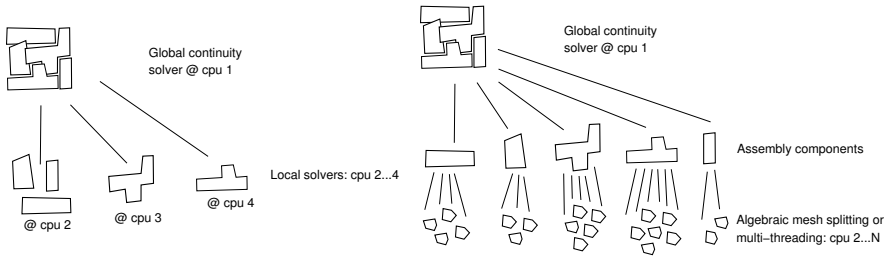


Fig. 3 Parallel implementation for small commodity clusters (left) and HPC systems (right).

Assume that one MPI process lives on each multi-core unit, and OpenMP parallelization occurs below, i.e. inside the multi-core NUMA unit [12, 13]. Actually, a good practice for computational performance is to set the number of OpenMP threads equal to the physical number of cores inside one NUMA node. Fig. 4 depicts the scalability results of a multi-threaded CG solver running on a Cray XE6 node (left) and SGI Altix UV 100 shared memory cluster. We observe almost linear speed-up, $\times 6$ and $\times 8$, respectively, for threads placed inside a single multi-core die.

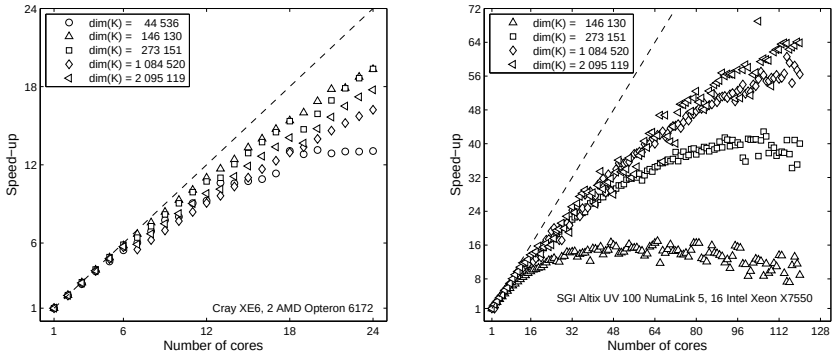


Fig. 4 Scalability results of a multi-threaded CG solver: Cray XE6 (left), SGI Altix UV (right).

3 Numerical Experiments

We consider the case of a linear elasticity. The model problem allows to describe the displacement $\mathbf{u} = (u_1, u_2, u_3)^t$ of an elastic body in its equilibrium position under the action of an external body force $\mathbf{f} = (f_1, f_2, f_3)^t$ and a surface charge $\mathbf{g}_N = (g_{N_1}, g_{N_2}, g_{N_3})^t$ distributed on $\partial\Omega_N$. Without getting technical about the spaces involved, i.e. the displacement weighting and trial solution W and V, for details see [15], the weak formulation for a problem of linear elasticity reads: find $\mathbf{u} \in V$ such that for all $\mathbf{w} \in W$

$$a(\mathbf{u}, \mathbf{w}) = F(\mathbf{w})$$

with

$$a(\mathbf{u}, \mathbf{w}) = \int_\Omega \lambda\, (\nabla \cdot \mathbf{u})(\nabla \cdot \mathbf{w})\, dx + \int_\Omega 2\,\mu\, \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{w})\, dx$$
$$F(\mathbf{w}) = \int_\Omega \mathbf{f} \cdot \mathbf{w}\, dx + \int_{\partial\Omega_N} \mathbf{g}_N \cdot \mathbf{w}\, ds$$

where λ and μ are the Lamé parameters, and $\varepsilon(\mathbf{u})$ is the infinitesimal strain tensor.

We have discretized the above problem using a $\mathbb{P}_2$ finite element. The FE model consists of three sub-domains, each triangulated independently, see Fig. 5. When working with fine meshes, the finest sub-domain contains roughly 3.6 million unknowns. We have used 4 computational nodes of a Cray XE6, with a total of 96 cores. The tasks were executed by 4 MPI processes each with 24 OpenMP threads, see Fig. 3 (right) (one MPI for a global continuity solver, one MPI per sub-domain). Three domain decomposition algorithms for non-matching meshes (Dirichlet-Neumann, Neumann-Neumann and FETI) have been implemented using a modified version of the integrated environment `FreeFem++` [14].

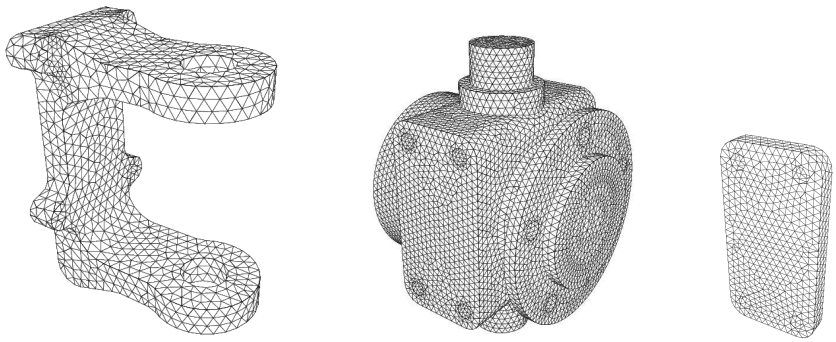


Fig. 5 A three component assembly. Non-matching triangulations are clear.

For simplicity reasons (to avoid floating sub-domains), we have set that each sub-domain has a part of its boundary belonging to a Dirichlet datum $\mathbf{u} = \mathbf{g}_D$ on $\partial\Omega_D$ (bolt holes of the left and right components, a back face of the middle component). All components are subject to the gravitational force. Portion of a front face of the right component is subject to a surface charge. We have set E=210 GPa, $\nu = 0.3$, E=105 GPa, $\nu = 0.34$ and E=117 GPa, $\nu = 0.33$ for the left, middle and right component, respectively; recall: $\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$. For each sub-domain, we have set the initial solution $\mathbf{u}_{ih}^{(0)} = (0,0,0)^t$.

On Fig. 6, we have visualized the computed displacements at iterations 1, 2 and 10; the computational time was 73 seconds per a single global iteration in the FETI method (fine meshes). The rate of convergence is shown in Fig. 7 for the Dirichlet-Neumann and Neumann-Neumann methods, respectively. The FETI method exhibits performance similar to the Neumann-Neumann method. The computations have been repeated for quasi-uniform coarse, medium and fine triangulations; for the mixed test we have used coarse, fine, medium triangulations for the left, middle and right component, respectively.

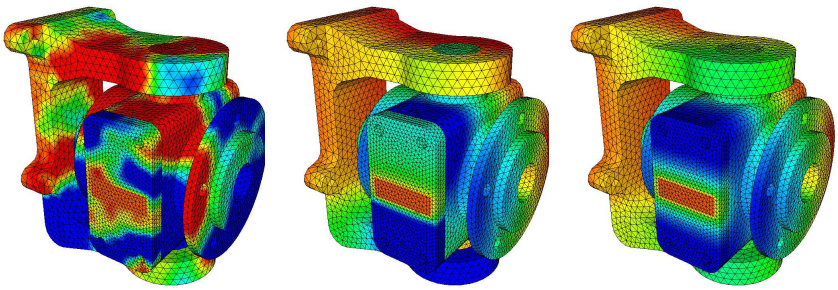


Fig. 6 The FETI method for non-matching triangulations. Lineal elasticity problem. Displacement field at iterations: 1, 2, 10.

4 Conclusions

We have introduced a comprehensive framework that allows to automate numerical simulations of multi-component CAD assemblies in the sense that meshes can be independently updated for each component. This paper has presented the CAD-based domain decomposition method. We have implemented the Dirichlet-Neumann, Neumann-Neumann and FETI methods for non-matching triangulations. Numerical results have indicated that all above methods are highly accurate finite element approximations for problems of linear elasticity. We have compared con-

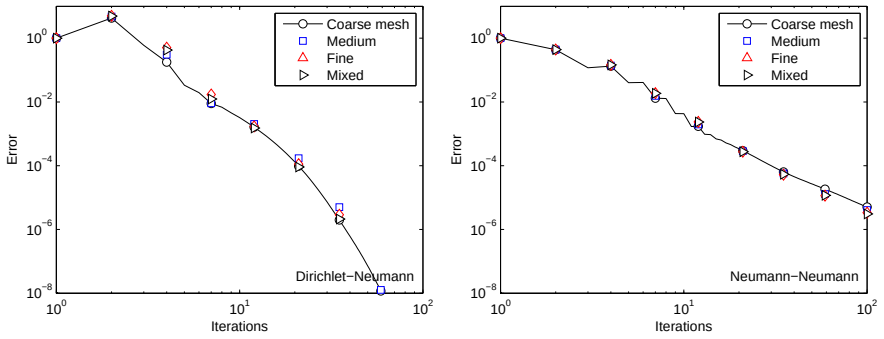


Fig. 7 Relative L^2 error for the Dirichlet-Neumann (left) and Neumann-Neumann (right) methods. The curves depict different levels of component mesh resolution H/h.

vergence properties of the three methods. The Dirichlet-Neumann method exhibits better convergence and is the most simple to implement.

References

1. D. Funaro, A. Quarteroni and P. Zanolli: An iterative procedure with interface relaxation for domain decomposition methods. SIAM J. Numer. Anal. **25(6)**, 1213–1236 (1988)
2. P. Le Tallec, Y.H. De Roeck and M. Vidrascu: Domain decomposition methods for large linearity elliptic three-dimensional problems. J. Comput. Appl. Math. **34(1)**, 93–117 (1991)
3. C. Farhat and F.X. Roux: A method of finite element tearing and interconnecting and its parallel solution algorithm. Internat. J. Numer. Methods Engrg. **32**, 1205–1227 (1991)
4. F. Hecht, J.L. Lions and O. Pironneau: Domain decomposition algorithm for computer aided design. Applied nonlinear analysis, 185–198 (2002)
5. P. Heintz and P. Hansbo: Stabilized Lagrange multiplier methods for bilateral elastic contact with friction. Comput. Methods Appl. Mech. Engrg. **195(33-36)**, 4323–4333 (2006)
6. S. Hartmann and E. Ramm: A mortar based contact formulation for non-linear dynamics using dual Lagrange multipliers. Finite Elem. Anal. Des. **44(5)**, 245–258 (2008)
7. L. Nilsson and J. Forsberg: Evaluation of response surface methodologies used in crashworthiness optimization. International journal of impact engineering **32**, 759–777 (2006).
8. D.J. Benson and J.O. Hallquist: A single surface contact algorithm for the post-buckling analysis of shell structures. Comput. Methods Appl. Mech. Engrg. **78(2)**, 141–163 (1990)
9. M.A. Puso and T.A. Laursen: A 3D contact smoothing method using Gregory patches. Internat. J. Numer. Methods Engrg. **54**, 1161–1194 (2002)
10. Parametric Technology Corporation: PRO/ENGINEER Wildfire 4.0 Part modeling. Help topic collection, (2008)
11. SIMULIA: Realistic Simulation Inside CATIA V5. Abaqus for CATIA V5, R19 (2008)
12. J.M. Bull, J. Enright, X. Guo, C. Maynard and F. Reid: Performance Evaluation of Mixed-Mode OpenMP/MPI Implementations. Int. J. Parallel Prog. **38(5-6)**, 396–417 (2010)
13. L. Smith and M. Bull: Development of mixed mode MPI / OpenMP applications. Sci. Program. **9(2-3)**, 83–98 (2001)
14. O. Pironneau, F. Hecht, A. Le Hyaric and K. Ohtsuka: FreFem++. URL `http://www.freefem.org/`
15. A. Ern: Aide-mémoire, Eléments finis Dunod, Paris (2005)

A finite volume Ventcell-Schwarz algorithm for advection-diffusion equations

Laurence Halpern[1] and Florence Hubert[2]

1 Introduction

Consider a two-dimensional domain Ω, and the boundary value problem

$$\mathscr{L}u := -\mathrm{div}(\nu(x)\nabla u) + \mathrm{div}(\mathbf{b}(x)u) + \eta(x)u = f, \tag{1}$$

with homogeneous boundary condition $u = 0$ on the boundary $\partial\Omega$. The Ventcell-Schwarz iterative method has been introduced in [9] for the resolution of (1) in parallel. A nonoverlapping decomposition of Ω into two subdomains Ω_j is given, with common boundary Γ. The algorithm defines a sequence of solutions u_j^n of equation (1) in Ω_j, related by two transmission conditions, for $(i,j) = (1,2)$ or $(2,1)$:

$$(\nu\partial_{n_j} - \frac{1}{2}\mathbf{b}\cdot\mathbf{n}_j + \Lambda)\, u_j^n = (-\nu\partial_{n_i} + \frac{1}{2}\mathbf{b}\cdot\mathbf{n}_i + \Lambda)\, u_i^{n-1} \text{ on } \Gamma.$$

The boundary operator Λ involves second order derivatives along the boundary. In the case where Γ is a vertical line, it can be written as $\Lambda\phi = p\phi - q\partial_y(\nu\partial_y\phi)$, with two real parameters p and q to be chosen adequately. By Lax-Milgram theorem, if $\nu \geq \nu_0 > 0$ and $\eta + \frac{1}{2}\mathrm{div}(\mathbf{b}) > 0$, the well-posedness of the boundary value problem is ensured as soon as p and q are positive. If $q = 0$, Λ reduces to Robin operator, first used in [10]. Numerical evidences with a finite element scheme were given in [9] that these transmission conditions outperform significantly the Robin-Schwarz algorithm . Further analysis has been conducted in [5] in a model case, where the coefficients p and q were obtained by optimization of the convergence factor of the algorithm, defined for two half planes, in the Fourier variables. Asymptotic values in terms of the discretization parameters were given (see Section 4).

The discrete counterpart of the algorithm in the Robin case $q = 0$ has been analyzed first in [1] and extended in [3] and [2] in the finite volume framework. For an analysis in the finite element context see [6]. The study of the Ventcell case $(p,q > 0)$ is, as far as we know, new. The scheme is fully described for the first time in this paper, and simulations are presented. The error analysis and the proofs of well-posedness and convergence will appear in an extended paper [7].

The first step, in section 2, is to write a finite volume scheme for the discretization of the subdomain problem. We use a two point flux approximation for the diffusive flux and a family of discrete convective fluxes as in [4], specially designed to handle the boundary condition. The discretisation of the boundary operator appearing in (1)

[1] LAGA, UMR 7539 CNRS, Université PARIS 13, 93430 VILLETANEUSE, FRANCE, e-mail: halpern@math.univ-paris13.fr .[2] Université Aix-Marseille, LATP, 39 rue F. Joliot Curie 13 453 Marseille Cedex 13, FRANCE e-mail: florence.hubert@univ-amu.fr

is performed. Non conforming meshes on the interface are considered as they can be useful for local refinement, see [8] for large scale computations.

The discrete Schwarz algorithm is described in section 3. In opposition to the Robin case, the convective flux on the interface has to be modified to get the convergence towards the approximation of (1) on Ω.

Finally, numerical examples illustrate the properties of the scheme , among which the improvement of the algorithm over the Robin algorithm.

2 Finite volume discretization for Ventcell transmission condition

We first introduce the necessary tools for finite volume design in the case of elliptic equation with mixed boundary conditions, Dirichlet on $\Gamma_D \subset \partial\Omega$ and possibly Ventcell (2) on $\Gamma \subset \partial\Omega$ (see [3] for the standard part of the notations).

Admissible Meshes Let Ω be an open polygonal set, $\mathfrak{M}$ a family of polygonal control volumes such that $\overline{\Omega} = \cup_{\mathrm{K}\in\mathfrak{M}}\mathrm{K}$, with $\mathrm{K}\cap\mathrm{L} = \emptyset$ if $\mathrm{K} \neq \mathrm{L}$. $\mathfrak{M}$ is an *admissible finite volume mesh* if there exists a family of points $(x_\mathrm{K})_{\mathrm{K}\in\mathfrak{M}}$ that satisfies $(x_\mathrm{K}, x_\mathrm{L}) \perp \sigma$ if $\sigma = \partial\mathrm{K}\cap\partial\mathrm{L}$. If all control volumes K are triangles, the family of circumcenters of the triangles satisfies this orthogonality condition. The set of all edges σ of control volumes is denoted by $\mathscr{E}$. It is divided into three sets: the edges located inside the domain Ω, $\mathscr{E}_{int} = \{\sigma \in \mathscr{E}/\sigma = \partial\mathrm{K}\cap\partial\mathrm{L}\}$, the edges $\mathscr{E}_D$ located on an external Dirichlet boundary Γ_D, and the edges $\mathscr{E}_\Gamma$ located on Γ. Finally, for any K in $\mathfrak{M}$, $\mathscr{E}_\mathrm{K}$ stands for the edges of its boundary $\partial\mathrm{K}$.

For any $\sigma \in \mathscr{E}_\mathrm{K}$, $\mathbf{n}_{\mathrm{K}\sigma}$ is the outward-pointing unit vector orthogonal to σ, $d_{\mathrm{K},\sigma} > 0$ the distance from x_K to σ, $d_\sigma = d_{\mathrm{K},\sigma}$ if $\sigma \in \mathscr{E}_D \cup \mathscr{E}_\Gamma$ and $d_\sigma = d_{\mathrm{K},\sigma} + d_{\mathrm{L},\sigma}$ is the distance between x_K and x_L if $\sigma = \partial\mathrm{K}\cap\partial\mathrm{L} \in \mathscr{E}_{int}$.

Let $|\mathscr{E}_\Gamma|$ be the cardinality of $\mathscr{E}_\Gamma$, the edges of $\mathscr{E}_\Gamma$ are reordered as $\{\sigma_i\}$, with $\sigma_i \cap \sigma_{i+1}$ reduced to a single point denoted by $x_{i+\frac{1}{2}}$. The control volume associated to σ_i is denoted by K_i.

For each $\mathrm{K} \in \mathfrak{M}$ or $\sigma \subset \Gamma$, $|\mathrm{K}|$ denotes the area of K, and $|\sigma|$ is the length of σ.

The complete admissible finite volume mesh for the boundary value problem is $\mathscr{T} = \mathfrak{M} \cup \mathscr{E}_\Gamma$. Figure 1 summarizes these notations.

Composite meshes The subdomains Ω_j are endowed with admissible meshes $\mathscr{T}_j = \mathfrak{M}_j \cup \mathscr{E}_\Gamma^j$, with two different meshes on Γ. The meshes $\mathscr{T}_1$ and $\mathscr{T}_2$ are said to be compatible if they coincide on Γ or equivalently if $\mathscr{E}_\Gamma^1 = \mathscr{E}_\Gamma^2$. We then define $\mathscr{E}_\Gamma = \mathscr{E}_\Gamma^1 = \mathscr{E}_\Gamma^2$. Any non compatible couple of meshes $(\mathscr{T}_1, \mathscr{T}_2)$ is made compatible by redefining the edges on Γ: in the example of Grid # 2 in Fig. 2, $\#\mathscr{E}_\mathrm{K} = 5$ for any control volume $\mathrm{K} \in \mathfrak{M}_1$ touching Γ. An edge of $\mathscr{E}_\Gamma$ is $\partial\mathrm{K}_1 \cap \partial\mathrm{K}_2$ with $\mathrm{K}_i \in \mathscr{T}_i$.

Finally a *composite mesh* associated to $\Omega = \Omega_1 \cup \Omega_2$ is a quadruplet $\mathscr{T} = (\mathfrak{M}, \mathfrak{M}_1, \mathfrak{M}_2, \mathscr{E}_\Gamma)$ such that each mesh $\mathfrak{M}_j$ is an admissible mesh for Ω_j, $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are compatible, and $\mathfrak{M} = \{K \in \mathfrak{M}_1 \cup \mathfrak{M}_2\}$.

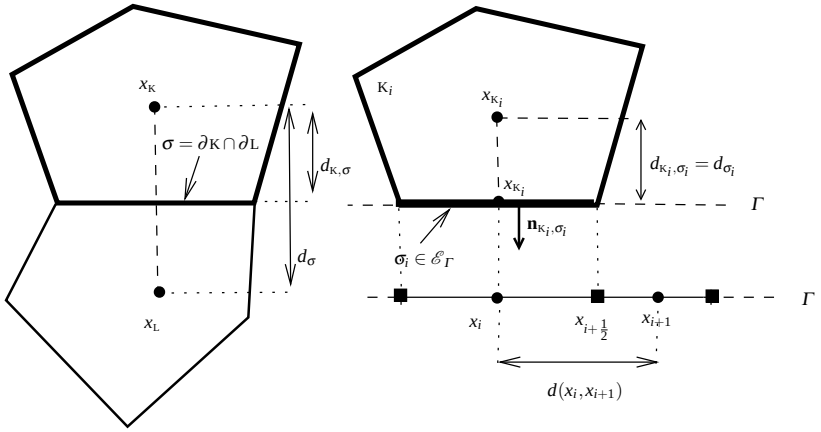


Fig. 1 Notations for an admissible mesh

A two-points flux approximation for Ventcell boundary conditions On each subdomain Ω_j, we approximate the problem $\mathscr{L}u_j = f$ with homogeneous Dirichlet boundary condition on $\Gamma_D^j = \partial\Omega_j \cap \partial\Omega$, and Ventcell boundary condition on $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$:

$$(\nu\partial_{n_j} - \frac{1}{2}\mathbf{b}\cdot\mathbf{n}_j + \Lambda)\, u_j = g_j. \tag{2}$$

For sake of clarity, the dependency on the index of the subdomain Ω_j will be omitted in this paragraph.

We introduce two sets $\mathbf{u}^{\mathfrak{M}} = (u_{\mathrm{K}})_{\mathrm{K}\in\mathfrak{M}}$ and $\mathbf{u}^{\mathscr{E}_\Gamma} = (u_\sigma)_{\sigma\in\mathscr{E}_\Gamma}$ of unknowns, one for the control volumes, one for the edges of the boundary $\mathscr{E}_\Gamma$. We define $\mathbf{u}^{\mathscr{T}} = (\mathbf{u}^{\mathfrak{M}}, \mathbf{u}^{\mathscr{E}_\Gamma})$. The discrete volume equations will be obtained, first by integrating the volume equation on a control volume K, second by integrating the boundary condition on the boundary control cell σ_i.

Equation on $\mathrm{K} \in \mathfrak{M}$

Integrating the equation (1) on the control volume K, we get:

$$\sum_{\sigma\in\mathscr{E}_{\mathrm{K}}} \left(-\int_\sigma \nu\nabla u\cdot\mathbf{n}_{\mathrm{K}\sigma}\, ds + \int_\sigma \mathbf{b}\cdot\mathbf{n}_{\mathrm{K}\sigma} u\, ds \right) + \int_{\mathrm{K}} \eta u\, dx = \int_{\mathrm{K}} f(x) dx.$$

The volume term $\int_{\mathrm{K}} \eta u\, dx$ can be approximated by $\eta_{\mathrm{K}} u_{\mathrm{K}}$ with $\eta_{\mathrm{K}} = \frac{1}{|\mathrm{K}|}\int_{\mathrm{K}} \eta$. The total flux in K is the sum on the edges of K of the diffusive fluxes $-\int_\sigma \nu\nabla u\cdot\mathbf{n}_{\mathrm{K}\sigma}\, ds$ and the convective fluxes $\int_\sigma \mathbf{b}\cdot\mathbf{n}_{\mathrm{K}\sigma} u\, ds$, that can be approximated respectively by the discrete fluxes $F^d_{\mathrm{K},\sigma}$, $F^c_{\mathrm{K},\sigma}$ to be defined below. Defining the total discrete flux on the edge σ as $F_{\mathrm{K},\sigma} = F^d_{\mathrm{K},\sigma} + F^c_{\mathrm{K},\sigma}$, the equation on $\mathrm{K} \in \mathfrak{M}$ can be approximated by

$$\forall \mathrm{K} \in \mathfrak{M}, \quad \sum_{\sigma\in\mathscr{E}_{\mathrm{K}}} F_{\mathrm{K},\sigma} + |\mathrm{K}|\eta_{\mathrm{K}} u_{\mathrm{K}} = \int_{\mathrm{K}} f(x)\, dx. \tag{3}$$

We use the classical diffusive discrete flux

$$F^d_{\mathrm{K},\sigma} = |\sigma|\nu_\sigma \frac{u_\mathrm{K} - \bar{u}_\sigma}{d_\sigma} \text{ with } \bar{u}_\sigma = \begin{cases} u_\mathrm{L} & \text{if } \sigma = \partial\mathrm{K} \cap \partial\mathrm{L} \in \mathscr{E}_{int}, \\ 0 & \text{if } \sigma \in \mathscr{E}_D, \\ u_\sigma & \text{if } \sigma \in \mathscr{E}_\Gamma, \end{cases} \tag{4}$$

with $\nu_\sigma = \frac{1}{|\sigma|}\int_\sigma \nu(s)\,ds$ or $\nu_\sigma = \nu(x_\sigma)$, ($x_\sigma$ center of σ) in the case of regular ν.

We introduce a general discrete convection flux in the form

$$F^c_{\mathrm{K},\sigma} = \frac{1}{2}|\sigma| b_{\mathrm{K}\sigma}(u_\mathrm{K} + \bar{u}_\sigma) + \frac{|\sigma|\nu_\sigma}{d_\sigma} B_\sigma\left(\frac{d_\sigma b_{\mathrm{K}\sigma}}{\nu_\sigma}\right)(u_\mathrm{K} - \bar{u}_\sigma), \tag{5}$$

where $b_{\mathrm{K}\sigma} = \frac{1}{|\sigma|}\int_\sigma \mathbf{b}\cdot\mathbf{n}_{\mathrm{K}\sigma}$, and for all edge σ, B_σ is an even Lipschitz continuous function such that

$$B_\sigma(0) = 0, \quad B_\sigma(s) + 1 > \underline{c} > 0 \text{ for } s \neq 0. \tag{6}$$

This frame, introduced in [4], includes the centered scheme $B_\sigma(s) := B^c(s) = 0$, the upwind scheme $B_\sigma(s) := B^{up}(s) = \frac{1}{2}|s|$, and the Scharfetter-Gummel scheme $B_\sigma(s) := B^{SG}(s) = \frac{1}{2}(\frac{s}{e^s-1} - \frac{s}{e^{-s}-1}) - 1$. Each of these approximations can be seen as a stabilization of the centered scheme. We will take advantage of this flexibility in the convergence analysis of the algorithm (see Theorem 2).

Equation for $\sigma \in \mathscr{E}_\Gamma$. Integrate the Ventcell boundary condition (2) on the edge $\sigma_i \in \mathscr{E}_\Gamma$ to obtain

$$\int_{\sigma_i} \nu\nabla u\cdot\mathbf{n}_{\mathrm{K}_i\sigma_i}\,ds - \frac{1}{2}\int_{\sigma_i}\mathbf{b}\cdot\mathbf{n}_{\mathrm{K}_i\sigma_i}u\,ds + p\int_{\sigma_i} u\,ds + q[-\nu\partial_y u]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} = \int_{\sigma_i} g(s)\,ds.$$

Define the discrete 1D flux $F_{i+\frac{1}{2}}$ as an approximation of $-\nu\frac{\partial u}{\partial y}(x_{i+\frac{1}{2}})$, given by

$$F_{i+\frac{1}{2}} = -\nu(x_{i+\frac{1}{2}})\frac{u_{\sigma_{i+1}} - u_{\sigma_i}}{d(x_{i+1}, x_i)} \quad \text{for } i = 0, \cdots, |\mathscr{E}_\Gamma|, \tag{7}$$

with the convention $u_{\sigma_0} = 0$ and $u_{\sigma_{|\mathscr{E}_\Gamma|+1}} = 0$. We obtain for all $\sigma \in \mathscr{E}_\Gamma$ the equation

$$-F_{\mathrm{K},\sigma} + \frac{1}{2}b_{\mathrm{K}\sigma}m_\sigma u_\sigma + (\Lambda^{\mathscr{E}_\Gamma}\mathbf{u}^{\mathscr{E}_\Gamma})_\sigma = \int_\sigma g(s)\,ds, \tag{8}$$

where the discrete boundary operator $\Lambda^{\mathscr{E}_\Gamma}$ is defined by

$$(\Lambda^{\mathscr{E}_\Gamma}\mathbf{u}^{\mathscr{E}_\Gamma})_\sigma = p|\sigma|u_\sigma - q(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}), \text{ for } \sigma = \sigma_i. \tag{9}$$

Properties of the scheme

By construction $\Lambda^{\mathscr{E}_\Gamma}$ is a symmetric and positive definite matrix. Therefore classical a priori estimates together with assumptions (6), induce the well-posedness of the scheme (3)-(8), see [7]. Furthermore, the scheme is of order 1.

3 A discrete Schwarz algorithm for Ventcell transmission conditions

Discrete Schwarz algorithm Given a composite mesh $\mathscr{T} = (\mathfrak{M}, \mathfrak{M}_1, \mathfrak{M}_2, \mathscr{E}_\Gamma)$, the discrete Schwarz algorithm consists, with suitable initial data, in finding for all $n \geq 1$, the solutions $\mathbf{u}^{\mathscr{T}_j,n} = (\mathbf{u}^{\mathfrak{M}_j,n}, \mathbf{u}^{\mathscr{E}_\Gamma^j,n})$ of the linear system

$$\forall \mathrm{K} \in \mathfrak{M}_j, \quad \sum_{\sigma \in \mathscr{E}_\mathrm{K}} (F_{\mathrm{K},\sigma})_j^n + |\mathrm{K}| \eta_\mathrm{K}(u_\mathrm{K})_j^n = \int_\mathrm{K} f(x)\,dx, \tag{10-a}$$

$$\begin{aligned} \forall \sigma \in \mathscr{E}_\Gamma, \quad & -(F_{\mathrm{K},\sigma})_j^n + \frac{1}{2}|\sigma| b_{\mathrm{K}_j,\sigma}(u_\sigma)_j^n + (\Lambda^{\mathscr{E}_\Gamma} u^{\mathscr{E}_\Gamma^j,n})_\sigma \\ & = (F_{\mathrm{K},\sigma})_i^{n-1} - \frac{1}{2}|\sigma| b_{\mathrm{K}_i,\sigma}(u_\sigma)_i^{n-1} + (\Lambda^{\mathscr{E}_\Gamma} u^{\mathscr{E}_\Gamma^i,n-1})_\sigma. \end{aligned} \tag{10-b}$$

Limit of the discrete Schwarz algorithm Assume that the algorithm (10) converges as n tends to infinity. The limit $\mathbf{u}^{\mathscr{T}_j,\infty} = (\mathbf{u}^{\mathfrak{M}_j,\infty}, \mathbf{u}^{\mathscr{E}_\Gamma^j,\infty})$ is solution of the scheme

$$\forall \mathrm{K} \in \mathfrak{M}_j, \quad \sum_{\sigma \in \mathscr{E}_\mathrm{K}} (F_{\mathrm{K},\sigma})_j^\infty + |\mathrm{K}| \eta_\mathrm{K}(u_\mathrm{K})_j^\infty = \int_\mathrm{K} f(x)\,dx, \tag{11-a}$$

$$\begin{aligned} \forall \sigma \in \mathscr{E}_\Gamma, \quad & -(F_{\mathrm{K},\sigma})_j^\infty + \frac{1}{2}|\sigma| b_{\mathrm{K}_j,\sigma}(u_\sigma)_j^\infty + (\Lambda^{\mathscr{E}_\Gamma^j} \mathbf{u}^{\mathscr{E}_\Gamma,\infty})_\sigma \\ & = (F_{\mathrm{K},\sigma})_i^\infty - \frac{1}{2}|\sigma| b_{\mathrm{K}_i,\sigma}(u_\sigma)_i^\infty + (\Lambda^{\mathscr{E}_\Gamma} \mathbf{u}^{\mathscr{E}_\Gamma^i,\infty})_\sigma. \end{aligned} \tag{11-b}$$

The expected limit However, we expect the convergence towards the classical two point flux finite volume scheme, associated to the mesh $\mathfrak{M}$ for the problem (1) on Ω, which consists in finding $\mathbf{u}^\mathfrak{M} = (u_\mathrm{K})_{\mathrm{K} \in \mathfrak{M}}$ solution of the discrete problem

$$\forall \mathrm{K} \in \mathfrak{M}, \quad \sum_{\sigma \in \mathscr{E}_\mathrm{K}} F_{\mathrm{K},\sigma} + |\mathrm{K}| \eta_\mathrm{K} u_\mathrm{K} = \int_\mathrm{K} f(x)\,dx. \tag{12}$$

If the composite mesh $\mathfrak{M}$ is non admissible in the neighborhood of Γ (Figure 2 right), the solution $u^\mathfrak{M}$ still approximates the solution u of (1), but with an error of order $\mathrm{size}(\mathfrak{M})^{\frac{1}{2}}$ only (See [3]).

The solutions of the schemes (12) and (11) can coincide only when the fluxes in (11) are modified, as stated in the next theorem.

Theorem 1. *Let* $\mathbf{u}^{\mathfrak{M}}$ *be the solution of* (12)*, with a convective flux in* (5) *defined by a function* B_σ*, satisfying*

$$B_\sigma(0)=0, \qquad B_\sigma(s) > -1+\frac{1}{2}|s|. \tag{13}$$

Define for $\sigma \in \mathscr{E}$ *the functions* $\bar{B}_\sigma$ *by*

$$\bar{B}_\sigma(s) = \begin{cases} B_\sigma(s) & \text{if } \sigma \notin \mathscr{E}_\Gamma, \\ \frac{1}{2}\left(1-B_\sigma(2s)\right) \pm \frac{1}{2}\sqrt{(1-s+B_\sigma(2s))(1+s+B_\sigma(2s))} & \text{if } \sigma \in \mathscr{E}_\Gamma. \end{cases} \tag{14}$$

Then, for this modified choice of fluxes $\bar{B}_\sigma$*, there exists* $\mathbf{u}^{\mathscr{T}_j,\infty} = (\mathbf{u}^{\mathfrak{M}_j,\infty}, \mathbf{u}^{\mathscr{E}_\Gamma,\infty})$ *for* $j=1,2$*, solution of* (11)*, and* $u_K^{\mathfrak{M}} = u_K^{\mathfrak{M}_j}$ *for* $K \in \mathfrak{M}_j$*.*

Proof. Let $u_{\mathrm{K}}^{\mathfrak{M}_j,\infty} = u_{\mathrm{K}}$ for all $\mathrm{K} \in \mathfrak{M}_j$. First for K such that $\mathscr{E}_{\mathrm{K}} \cap \mathscr{E}_\Gamma = \emptyset$, equation (11-a) is nothing but equation (12). However, the construction of the edge unknowns $\mathbf{u}^{\mathscr{E}_\Gamma^j}$ requires some care.

For $\sigma \in \mathscr{E}_\Gamma$, equation (11-b) written for $(j,i)=(1,2)$ and $(2,1)$ yields

$$\Lambda^{\mathscr{E}_\Gamma} u^{\mathscr{E}_\Gamma^1,\infty} = \Lambda^{\mathscr{E}_\Gamma} u^{\mathscr{E}_\Gamma^2,\infty}.$$

Thus, using the invertibility of $\Lambda^{\mathscr{E}_\Gamma}$, we obtain that $u^{\mathscr{E}_\Gamma^1,\infty} = u^{\mathscr{E}_\Gamma^2,\infty} = u^{\mathscr{E}_\Gamma,\infty}$ and $(F_{\mathrm{K},\sigma})_1^\infty = -(F_{\mathrm{K},\sigma})_2^\infty$. Finally equation (11-a) coincides with equation (12) if

$$F_{\mathrm{K},\sigma} = (F_{\mathrm{K},\sigma})_1^\infty. \tag{15}$$

Define $d_{\mathrm{K}_1\sigma}$, $d_{\mathrm{K}_2\sigma}$ and s by $s = \frac{b_{\mathrm{K}_1\sigma} d_{\mathrm{K}_1\sigma}}{\nu_{\mathrm{K}_1\sigma}} = -\frac{b_{\mathrm{K}_2\sigma} d_{\mathrm{K}_2\sigma}}{\nu_{\mathrm{K}_2\sigma}} = \frac{b_{\mathrm{K}_1\sigma} d_{\mathrm{K}_1\mathrm{K}_2}}{2\nu_\sigma}$. We then have for j=1,2

$$(F_{\mathrm{K},\sigma})_j^\infty = \frac{|\sigma|\nu_{\mathrm{K}_j\sigma}}{d_{\mathrm{K}_j\sigma}}(u_{\mathrm{K}_j}^\infty - u_\sigma^\infty)(1+\bar{B}_\sigma(s)) + \frac{1}{2}|\sigma| b_{\mathrm{K}_j\sigma}(u_{\mathrm{K}_j}^\infty + u_\sigma^\infty).$$

Identifying $(F_{\mathrm{K},\sigma})_1^\infty$ to $-(F_{\mathrm{K},\sigma})_2^\infty$ defines u_σ^∞, then (15) is equivalent to

$$B_\sigma(2s) = \bar{B}_\sigma(s) + \frac{1}{4}s^2(1+\bar{B}_\sigma(s))^{-1}. \tag{16}$$

Hence, to express $\bar{B}_\sigma(s)$ in terms of $B_\sigma(s)$, we solve the equation $X^2 + (1 - B_\sigma(2s))X + \left(\frac{1}{4}s^2 - B_\sigma(2s)\right) = 0$, Under condition (13), there exists a unique solution satisfying $\bar{B}_\sigma(0) = 0$, which is given in (14).

In this case, any solution of (11) is a solution of (12), which has a unique solution. □

Remark 1. Assumption (13) is satisfied by the upwind scheme, the Scharfetter-Gummel scheme and the centered scheme if $|s| < 1$. In the case of the Scharfetter-Gummel scheme, $\bar{B}_\sigma = B_\sigma$.

Convergence of the Schwarz algorithm

Theorem 2. *Let $\mathscr{T}_j$ be two compatible meshes of Ω_j, $j=1,2$ and $\mathscr{T}$ the associated composite mesh. With the assumptions in Theorem 1, the solution $(\mathbf{u}^{\mathfrak{M}_{j,n}})_{j=1,2}$ of the discrete Schwarz algorithm* (10) *converges to $\mathbf{u}^{\mathfrak{M}}$ solution of* (12) *as n tends to infinity.*

Hint on the proof. The proof is too long to be developed here, and will appear in [7]. By Theorem 1 the convergence of the Schwarz algorithm is equivalent to the convergence to 0 of the solution $\mathbf{u}^{\mathscr{T}_j,n}=(\mathbf{u}^{\mathfrak{M}_{j,n}},\mathbf{u}^{\mathscr{E}_\Gamma^j,n})$ of (10) when f is identically zero. That convergence is then obtained by an extension of P.L. Lions trick in [10], using the fact that $\Lambda^{\mathscr{E}_\Gamma}$ is a symmetric positive definite matrix.

4 Numerical experiments

The domain $\Omega=]-1,1[\times]0,1[$ is split into $\Omega_1=]-1,0[\times]0,1[$ and $\Omega_2=]0,1[\times]0,1[$ with an interface Γ at $x=0$. We compare the convergence behaviour of the optimized Schwarz algorithm for Robin and Ventcell transmission conditions. Define the mesh size on the interface, $h=\min(\max(|\sigma|,\ \sigma\in\mathscr{E}_j),\ j=1,2)$. Asymptotically optimal parameters (for small h) are taken from [5]. They have been determined to produce the smallest convergence factor over all frequencies supported by the grid.

$$\text{Robin}: \quad p^\star=\tfrac{h^{-\frac12}}{2}\sqrt{2\pi\nu(b_x^2+4\nu\eta)^{\frac12}}, \quad q^\star=0.$$

$$\text{Ventcell}: \quad p^\star=\tfrac{h^{-\frac14}}{2}\sqrt[4]{\tfrac{\nu\pi(b_x^2+4\nu\eta)^{\frac32}}{2}}, \qquad q^*=\frac{h^{\frac34}}{2}\sqrt[4]{\frac{8\nu}{\pi^3}(b_x^2+4\nu\eta)^{-\frac12}}.$$

The corresponding theoretical convergence factor of the algorithm (i.e. the factor of reduction of the L^2 norm of the error in one iteration) is

$$\text{Robin}: 1-\mathscr{O}(h^{\frac12}), \quad \text{Ventcell}: 1-\mathscr{O}(h^{\frac14}),$$

showing an improvement from Robin to Ventcell, since it is less dependent of the size of the mesh.

We choose $\nu=0.1$, $\overrightarrow{b}=(1,1)^t$, $\eta=1$. The source f is such that the exact solution of (1) is $u(x,y)=\sin(3\pi x)\sin(3\pi y)$. The Scharfetter-Gummel scheme is used for all edges. The algorithm is initialized with random data $(\mathbf{u}^{\mathfrak{M}_{j,0}})_{j=1,2}$. We illustrate our results on two families of grids presented in Figure 2, one is conforming (Grid # 1), the other non conforming (Grid # 2) at the interface Γ. We draw the convergence history for increasing mesh refinement, given by $i=3,4,5,6$. We stopped the algorithm as soon as $\left(\sum_{j=1,2}\|\mathbf{u}^{\mathfrak{M}_{j,n+1}}-\mathbf{u}^{\mathfrak{M}_{j,n}}\|^2_{L^2(\Omega_j)}\right)^{\frac12}\le 10^{-7}$. We can see the drastic improvement obtained by using the second order transmission condition, for which the convergence lines seem almost independent of h. The numerical con-

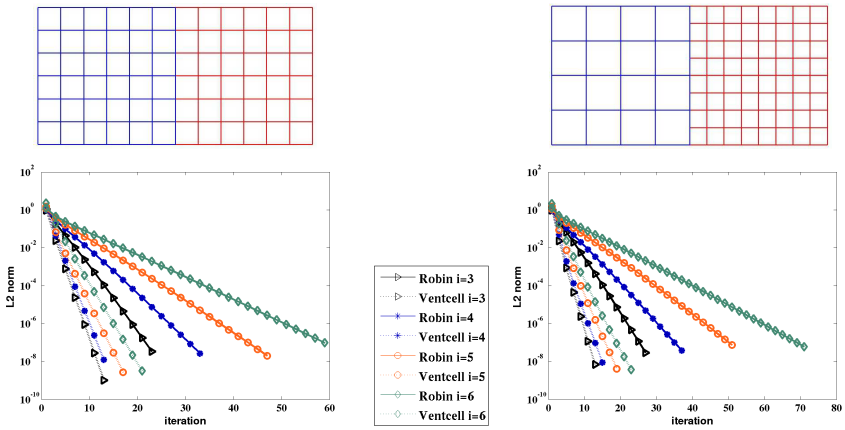


Fig. 2 (Left) A $6*2^i \times 6*2^i$ square grid on both Ω_1 and Ω_2. (Right) A $4*2^i \times 4*2^i$ grid on Ω_1 and a $8*2^i \times 8*2^i$ grid on Ω_2. Robin vs Ventcell. L^2 norm error w.r.t. iterations for increasing mesh refinements.

vergence factor behaves in $1-\mathcal{O}(h^\alpha)$ with $\alpha = 0.43$ for Robin-Grid # 1, $\alpha = 0.44$ for Robin-Grid # 2, $\alpha = 0.17$ for Ventcell-Grid # 1, $\alpha = 0.19$ for Ventcell-Grid # 2.

References

1. Achdou, Y., Japhet, C., Nataf, F., Maday, Y.: A new cement to glue non-conforming grids with Robin interface conditions: The finite volume case. Numer. Math. **92**(4), 593–620 (2002)
2. Boyer, F., Hubert, F., Krell, S.: Non-overlapping Schwarz algorithm for solving 2D m-DDFV schemes. IMA J. Numer. Anal. **30**, 1062–1100 (2009)
3. Cautrès, R., Herbin, R., Hubert, F.: The Lions domain decomposition algorithm on non-matching cell-centred finite volume meshes. IMA J. Numer. Anal. **24**(3), 465–490 (2004)
4. Chainais-Hillairet, C., Droniou, J.: Finite volume schemes for non-coercive elliptic problems with Neumann boundary conditions. IMA Journal of Numerical Analysis **31**(1), 61–85 (2011)
5. Dubois, O.: Optimized Schwarz methods for the advection-diffusion equation and for problems with discontinuous coefficients. Ph.D. thesis, McGill University, Canada (2007)
6. Gander, M.J., Japhet, C., Maday, Y., Nataf, F.: A new cement to glue nonconforming grids with Robin interface conditions: the finite element case. Lect. Notes Comp. Sci. Eng. **40**, 259–266 (2005)
7. Halpern, L., Hubert., F.: Optimized Schwarz algorithms in the classical finite volume framework. http://www.math.univ-paris13.fr/ halpern/Publis/HH.pdf (2013)
8. Halpern, L., Japhet., C., Szeftel, J.: Optimized Schwarz waveform relaxation and discontinuous Galerkin time stepping for heterogeneous problem. SIAM Journal on Numerical Analysis **50**(5), 2588–2611 (2012)
9. Japhet, C.: Méthode de décomposition de domaine et conditions aux limites artificielles en mécanique des fluides : méthode Optimisée d'Ordre 2. Ph.D. thesis, Université Paris 13, France (1998)
10. Lions, P.L.: On the Schwarz alternating method. III. A variant for nonoverlapping subdomains. Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989), pp. 202–223.

Domain Decomposition with Nesterov's Method

Firmin Andzembe[1], Jonas Koko[1], and Taoufik Sassi[2]

1 Introduction

Nesterov's method is a first order convex minimization method with convergence rate $O(1/k^2)$, see e.g. [4, 3]. The method can be used with either smooth or nonsmooth convex optimization problems. For constrained minimization, if the projection onto the constraints set is easy to compute, a projected gradient variant of the Nesterov method can be derived, see e.g. [1, 7].

In this paper we apply Nesterov's method to the domain decomposition. The model problem is the Poisson equation. As a first order optimization method, the Nesterov method needs, per iteration, only matrix/vector multiplcations while standard domain decomposition methods need matrices inversion through solution to linear systems, see e.g. [5, 6]. The Nesterov method is therefore well-suited for *Graphics Processing Unit* (GPU) architecture for which the (direct of iterative) linear solvers using complete or incomplete factorizations are inefficient, see, e.g., [2]. Moreover, the Nesterov method can be (theroetically) used for domain decomposition of nonsmooth problems (i.e. problems with L^1 terms)

The paper is organized as follows. In the next section we recall the Nesterov method for convex programming problem. The model (Poisson) problem and the domain decomposition are presented in Section 3. The Nesterov domain decomposition method is presented in Section 4 followed by preliminary numerical experiments in Section 5.

2 Nesterov's Method

Let F be a convex function defined on a finite dimensional space X. The subgradient of F at x is defined by

$$\partial F(x) = \{p \in X \mid F(y) \geq F(x) + (p, y-x),\ \forall y \in \mathrm{dom} F\}.$$

If F is differentiable, then $\partial F(x) = \{\nabla F(x)\}$.

Let $\delta > 0$ and assume that F is convex, lower-semicontinous function on X. It is easy to show that the problem

[1]LIMOS, Université Blaise Pascal – CNRS UMR 6158, F-63000 Clermont-Ferrand, France,e-mail: {andzembe}{koko}@isima.fr .[2] LMNO, Université de Caen – CNRS UMR, F-14032 Caen, France e-mail: sassi@univ-caen.fr

$$\min_y \delta F(y) + \frac{1}{2} \| y - x \|^2$$

always has a unique solution, verifying the equation

$$\delta \partial F(y) + y - x \ni 0$$

that is, formally

$$y = (I + \delta \partial F)^{-1}(x).$$

The mapping $(I + \delta \partial F)^{-1}$, called "proximal map of δF", is well defined and uniquely defined. If K is a closed and convex set and $F = \mathbf{1}_K$ (i.e. F is the characteristic function of K), then $(I + \delta \partial F)^{-1}$ is a projection onto K.

Consider the following optimization problem

$$\min_x \Phi(x) = F(x) + G(x), \tag{1}$$

where we assume that

- F is $\mathscr{C}^{1,1}$, i.e. the gradient ∇F is Lipschitz with some constant L;
- G is "simple" in the sens that the "prox" operator $(I + \delta \partial G)^{-1}$ is easy to compute (e.g. projection)

The most straightforward Nesterov method is the projected gradient (Beck and teboulle [1]), an adaptation of the gradient descent algorithm due to Nesterov [4]. The projected gradient method is outline in Algorithm 2. The rate of convergence of Algorithm 2 is given by the following theorem due to Beck and Teboulle [1].

Theorem 1. *Let $\{x^k\}$ be the sequence generated by Algorithm 2 with $\delta = 1/L$. Then*

$$\Phi(x^k) - \Phi(x^*) \leq \frac{L}{2k} \| x^0 - x^* \|^2,$$

for any $k \geq 1$ and for any x^* solution of the minimization problem (1).

Algorithm 2 Nesterov's projected gradient algorithm

(i) $k = 0$. Choose x^0 and $\delta > 0$

(ii) $k \geq 0$. Compute $x^{k+1} = (I + \delta \partial G)^{-1}(x^k - \delta \nabla F(x^k))$

To overcome the slow rate of convergence of Algorithm 2, Nesterov proposes in [3] an acceleration variant of the gradient descent. For solving minimization problems of the form (1), Beck and Teboulle propose Algorithm 3, variant of the Nesterov accelerated algorithm.

The rate of convergence of Algorithm 3 is given by the following theorem due to [1].

Algorithm 3 Accelerated Nesterov's Algorithm

$k=0$ $x^0, y^1 = x^0, t_1 = 1, \delta > 0$

$k \geq 0$ Compute x^k and y^k as follows

(i) $z^k = y^k - \delta \nabla F(y^k)$

(ii) $x^k = (I + \delta \partial G)^{-1}(z^k)$

(iii) $t_{k+1} = \frac{1}{2}\left(1 + \sqrt{1+4t_k^2}\right)$

(iv) $y^{k+1} = x^k + (t_k - 1)(x^k - x^{k-1})/t_{k+1}$

Theorem 2. *For any minimizer x^* of (1), the sequence $\{x^k\}$ generated by Algorithm 3 with $\delta = 1/L$ is such that*

$$\Phi(x^k) - \Phi(x^*) \leq \frac{2L}{(k+1)^2} \| x^0 - x^* \|^2, \tag{2}$$

for any $k \geq 1$.

3 Model problem and Domain Decomposition

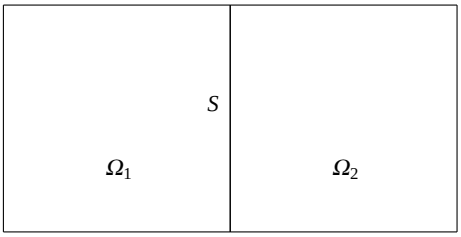


Fig. 1 Domain decomposition of Ω into two subdomains with S as the common interface

3.1 Model problem

Let Ω be a bounded domain in $\mathbb{R}^d$ ($d = 2, 3$) with Lipschitz-continuous boundary Γ. We consider in Ω the Poisson problem

$$-\Delta u = f, \quad \text{in } \Omega, \tag{3}$$

$$u = 0 \quad \text{on } \Gamma. \tag{4}$$

Setting

$$V = H_0^1(\Omega), \quad f(v) = \int_\Omega f v \mathrm{d}x \quad \text{and } a(v,v) = \int_\Omega \nabla v \cdot \nabla v \mathrm{d}x,$$

the Poisson problem (3)-(4) can be reformulated as the follwing convex minimization problem

$$\min_{v \in V} J(v) = \frac{1}{2} a(v,v) - f(v). \tag{5}$$

3.2 Domain decomposition

Let $\{\Omega_1,\ \Omega_2\}$ be a partition of Ω, as shown in Figure 1, and let $S = \partial\bar{\Omega}_1 \cap \partial\bar{\Omega}_2$, $v_i = v|_{\Omega_i}$ and

$$\Gamma_i = \Gamma \cap \partial\Omega_i, \quad V_i = \left\{ v \in H^1(\Omega_i),\ \ v|_{\Gamma_i} = 0 \right\}.$$

It follows that

$$a(v,v) = \sum_{i=1}^{2} a_i(v_i, v_i), \quad f(v) = \sum_{i=1}^{2} f_i(v_i), \quad J(v) = \sum_{i=1}^{2} J_i(v_i)$$

and the minimization problem (5) becomes

$$\min_{(v_1,v_2)} J_1(v_1) + J_2(v_2) \tag{6}$$

$$[v] := (v_1 - v_2)|_S = 0 \text{ on } S. \tag{7}$$

With the formulation (6)-(7), the continuity of the normal derivative accross S is ensured (implicitly) by the Lagrange multiplier associated with (7). Indeed, if (u_1, u_2) is the solution of the constrained optimization problem (6)-(7), then there exists $\lambda \in L^2(S)$ such that

$$\begin{aligned} a_i(u_i, v_i) &= f_i(v_i) + (-1)^i (\lambda, v_i)_S, \quad \forall v_i \in V_i,\ i = 1,2 \\ (\mu, [u])_S &= 0, \quad \forall \mu \in L^2(S), \end{aligned}$$

or

$$-\Delta u_i = f_i \text{ in } \Omega_i \quad \text{and} \quad \frac{\partial u_i}{\partial n_i} = (-1)^i \lambda \text{ on } S$$

so that

$$\lambda = -\frac{\partial u_1}{\partial n_1} = \frac{\partial u_2}{\partial n_2}. \tag{8}$$

3.3 Finite dimensional problem

Finite element or finite difference aproximations of the above Poisson problem leads to the quadratic forms

$$J_i(v_i) = \frac{1}{2} v_i^T A_i v_i - f_i^T v_i, \quad i = 1,2.$$

were A_i are symmetric positive definite matrices. For v_i we use the following decomposition

$$v_i = \begin{bmatrix} v_{iI} \\ v_{iS} \end{bmatrix}$$

where $v_{iS} = v_{i|S}$ (the subvector of interface unknowns) and $v_{iI} = v_{i|(\Omega \setminus S)}$ (the subvector of interior unknowns). Let us introduce the set K, defining the continuity condition

$$K = \{(v_1, v_2) \,:\, [v] = v_{1S} - v_{2S} = 0\}.$$

It is obvious that K is closed and convex. The finite dimensional constrained optimization problem is therefore

$$\min_{(v_1,v_2)\in K} J(v_1, v_2) = \sum_{i=1}^{2} J_i(v_i). \tag{9}$$

4 Nesterov domain decomposition method

Let us introduce the functions

$$F(v) = J_1(v_1) + J_2(v_2)$$
$$G(v) = 1_K(v).$$

G is the characteristic function of K. The finite dimensional (constrained) minimization problem (9) can be rewritten as the following convex unconstrained minimization problem

$$\min_v F(v) + G(v) \tag{10}$$

Note that F is a convex function and G is a charactersitic function of a closed convex set. Then the proximal map $(I + \delta \partial G)^{-1}$ is easy to compute. Indeed, for $p = (p_1, p_2)$

$$(I + \delta \partial G)^{-1}(p) = \arg\min_q \frac{1}{2} \parallel q - p \parallel^2 + \delta G(q) = (\tilde{p}_1, \tilde{p}_2)$$

where

$$\tilde{p}_i = \begin{bmatrix} p_{iI} \\ \frac{1}{2}(p_{1|S} + p_{2|S}) \end{bmatrix}, \quad i = 1,2, \tag{11}$$

the projection of (p_1, p_2) onto to K. The minimization problem can then be solved by the Nesterov Algorithm 3. The resulting domain decomposition method is described in Algorithm 4. The parallelizability of the method is obvious.

Algorithm 4 Nesterov domain decomposition algorithm

$k = 0$: u_i^0, $q_i^1 = u_i^0$, $t_1 = 1$, $\delta = 1/L$

$k \geq 0$: Compute u^k and q^{k+1} as follows

(i) $z_i^k = q_i^k - \delta(A_i q_i^k - b_i)$, $i = 1,2$

(ii) $u_i^k = \begin{bmatrix} z_{iI}^k \\ (z_{1S}^k + z_{2S}^k)/2 \end{bmatrix}$, $i = 1,2$

(iii) $t_{k+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right)$

(iv) $q_i^{k+1} = u_i^k + (t_k - 1)(u_i^k - u_i^{k-1})/t_{k+1}$, $i = 1,2$.

Since the domain decomposition is an optimization based, the jumps in a coefficient is not an issue. If in (3), the Laplacian operator is replaced by $\nabla \cdot (\alpha(x)\nabla u(x))$, then the continuity condition, i.e. (11), does not change while (8) becomes

$$\lambda = -\alpha_1 \frac{\partial u_1}{\partial n_1} = \alpha_2 \frac{\partial u_2}{\partial n_2},$$

assuming $\alpha_i = \alpha_{|\Omega_i}$, $i = 1,2$.

In the case of a decomposition with intersection of more than two subdomains, a special procedure must be carried out to ensure the continuity condition (11). For instance, in the case of an intersection of four subdomains, with $\{p_{iS}\}_{i=1,\ldots,4}$ the value of p at the corner of each subdomain, we must have

$$p_{2S} - p_{1S} = 0, \quad p_{3S} - p_{2S} = 0, \quad p_{4S} - p_{3S} = 0.$$

A straightforward calculation (using optimality conditions) yields

$$\tilde{p}_{iS} = \frac{1}{4} \sum_{\ell=1}^{4} p_{\ell S}, \quad i = 1, \ldots, 4.$$

5 Numerical experiments

The domain decomposition algorithm presented in the previous sections was implemented in Fortran 90, on a Linux cluster, using an MPI library. We use P^1 finite element method for the discretization. The Lipschitz constant L is approximated in the initialization step using the power method. Indeed, for the model problem

$L = \rho(A)$, the spectral radius of the Laplacian matrix. The stopping criterion is $(J(u^k) - J(u^{k-1}))/h < 10^{-6}$ where h is the size of the mesh.

We consider the domain $\Omega = (0,\ 1) \times (0,\ 1)$ and the right-hand side in (3) is adjusted such that the exact solution is $u(x,y) = (x-1)y\sin(x)\cos(2\pi y)$. Table 1 shows the number of iterations and CPU times (in seconds) for several mesh sizes and number of sub-domains. The CPU times given include the aproximation of L by the power method. We notice that, for the largest problem ($h = 1/256$), the standard speed-up (i.e. the number of degrees of freedom is constant while the number of sub-domains varies) obtained with the projected gradient Algorithm 4 is significant: about 43 for 32 sub-domains.

In Table 2 we report the results for the scaled speed-up, i.e. the number of sub-domains varies while the number of nodes in each sub-domain is kept fixed to 100×100 (10000 degrees of freedom). We notice that the number of iterations increases with the number of sub-domains: the number of iterations is multiplied by about 4 while the number of subdomains is multiplied by 36.

N_{SD}	$h = 1/16$	$h = 1/32$	$h = 1/64$	$h = 1/128$	$h = 1/256$
	IT/CPU	IT/CPU	IT/CPU	IT/CPU	IT/CPU
1	134/0.01	270/0.14	284/1.11	416/5.57	834/45.78
2	40/0.00	79/0.08	154/0.21	309/2.11	611/26.77
4	78/0.00	109/0.08	159/0.30	312/1.28	613/6.07
16	122/0.03	300/0.18	361/0.20	320/0.26	847/2.10
32	165/6.056	310/6.12	369/0.15	595/0.38	637/1.05

Table 1 Standard speed-up: N_{SD} := number of subdomains; h := mesh size; IT:= number of iterations; CPU:= CPU times in seconds.

N_{SD}	1	4	9	16	25	36
IT	440	486	730	974	1218	1461
CPU	2.84	3.28	5.32	14.40	7.55	8.76

Table 2 Scaled speed-up with 100×100 nodes in each sub-domain: N_{SD} := number of subdomains; IT:= number of iterations; CPU:= CPU times in seconds

6 Conclusion

A Nesterov domain decomposition algorithm for the Poisson problem has been introduced. The continuity condition on the interface is enforced using projection.

This approach is easy to implement and preliminary numerical experiments show that a significant speed-up is obtained. Nevertheless, it leads to a h-dependent algorithm. Further work is under way to improve the algorithm (preconditioning, restarting strategy, etc.)

References

1. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
2. Helfenstein, R., Koko, J.: Parallel preconditioned conjugate gradient algorithm on gpu. J. Computational Applied Mathematics **236**, 3584–3590 (2012)
3. Nesterov, Y.: A method for solving the convex programming problem with convergence rate $0(1/k^2)$. Dokl. Akad. Nauk. SSSR **269**(3), 543–547 (1983)
4. Nesterov, Y.: Smooth minimization of non-smooth functions. Mathematical programming (A) **103**, 127–152 (2005)
5. Quarteroni, A., Valli, A.: Domain Decomposition Methods for Partial Differential Equations. Oxford University Press (1999)
6. Toselli, A., Widlund, O.: Domain Decomposition Methods – Algorithms and Theory. Springer (2005)
7. Weiss P.; Blanc-Ferraud, L., Aubert, G.: Efficient schemes for total variation minimization under constraints in image processing. SIAM J. Scientific Computing **31**, 2047–2080 (2009)

Total-FETI method for solving contact elasto-plastic problems

Martin Cermak[1] and Stanislav Sysala[2]

1 Introduction

Contact problems with elasto-plastic bodies can be solved for example by primal-dual active set strategy, see e.g. [12]. In this paper, we propose a numerical method that combines the semi-smooth Newton method with the Total-FETI (TFETI) domain decomposition method and SMALSE method [1].

We consider a frictionless contact boundary condition between two bodies denoted as $\Omega^1, \Omega^2 \subset \mathbb{R}^3$, see Fig. 1. We assume that the bodies are fixed on the parts $\Gamma_U^1, \Gamma_U^2 \neq \emptyset$ of the boundaries. The load is represented by surface (prescribed on the boundaries parts Γ_N^1, Γ_N^2) and volume forces. The material of the bodies is described by the elasto-plastic constitutive model with the von Mises yield criterion and linear isotropic hardening [10]. For the sake of simplicity, we confine ourselves on one-step problem formulated in displacement. It leads to a minimization of the convex and smooth functional on a convex set. However the stress-strain relation is not smooth.

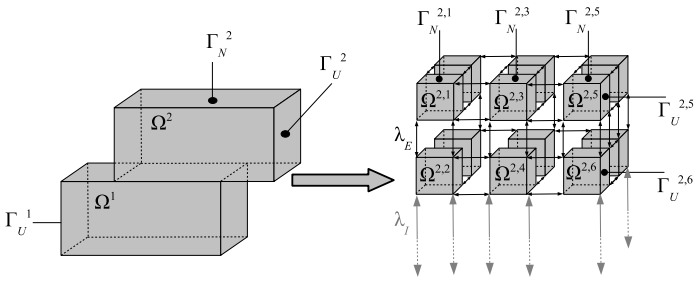


Fig. 1 Scheme of the geometry and domain decomposition

The problem is approximated by the finite element method. The finite element partition will be denoted as $\mathscr{T}_h = \mathscr{T}_h^1 \cup \mathscr{T}_h^2$ and consists of simplicial elements. In particular, displacement fields are approximated by continuous, piecewise linear functions and strain (stress) fields are approximated by piecewise constant functions. We will not investigate in detail the influence of domain and load approximation.

[1] IT4Innovations, VSB-TU Ostrava, 17. listopadu 15/2172, Ostrava-Poruba, 708 33, Czech Republic e-mail: martin.cermak@vsb.cz .[2] Institute of Geonics AS CR, v.v.i., Studentska 1768, Ostrava-Poruba, 708 00, Czech Republic, e-mail: stanislav.sysala@ugn.cas.cz

Since we will apply the TFETI domain decomposition method [2], we tear the bodies from the part of the boundary with the Dirichlet boundary condition, decompose it into subdomains, assign each subdomain by a unique number, and introduce new "gluing" conditions on the artificial intersubdomain boundaries and on the boundaries with imposed Dirichlet condition. In particular, the domain $\Omega_h^i \equiv \Omega^i$ is decomposed into a system of s_i disjoint polyhedral subdomains $\Omega^{i,p} \subset \Omega^i$, $p = 1,2,\ldots,s_i$, $i = 1,2$, see Fig. 1. The partition is conforming with the finite element partition $\mathscr{T}_h$.

The discretized problem can be classified as an optimization problem with simple equality and inequality contraints. In Section 2, we introduce and describe an algebraic formulation of the problem. We use the semi-smooth Newton method to approximate a non-quadratic functional by a quadratic one, see Section 3. The corresponding problem of quadratic programming is solved by the Total-FETI domain decomposition method in combination with SMALSE method, see Section 4. The elasto-plastic problem with contact was implemented into the MatSol library [8]. We illustrate the performance of our algorithm on a 3D benchmark problem in Section 5.

2 Algebraic formulation of the contact problem for elasto-plastic bodies

Algebraic formulation of the problem will be related to the domain decomposition. It means that a displacement vector $\mathbf{v} \in \mathbb{R}^n$ has the following structure:

$$\mathbf{v} = \left(\mathbf{v}_{1,1}^T, \mathbf{v}_{1,2}^T, \ldots, \mathbf{v}_{1,s_1}^T, \mathbf{v}_{2,1}^T, \ldots, \mathbf{v}_{2,s_2}^T\right)^T,$$

where $\mathbf{v}_{i,p}$ denotes the displacement vector on $\Omega^{i,p}$, $i = 1,2$. We define the space

$$\mathscr{V} := \{\mathbf{v} \in \mathbb{R}^n \mid \mathbf{B}_E \mathbf{v} = \mathbf{o}\}, \tag{1}$$

and the set of admissible displacement

$$\mathscr{K} := \{\mathbf{v} \in \mathbb{R}^n \mid \mathbf{B}_E \mathbf{v} = \mathbf{o},\ \mathbf{B}_I \mathbf{v} \leq \mathbf{c}_I\}. \tag{2}$$

Here the equality constraint matrix $\mathbf{B}_E \in \mathbb{R}^{m_E \times n}$ represents the gluing conditions among neighbouring subdomains and the Dirichlet boundary conditions. The inequality constraint matrix $\mathbf{B}_I \in \mathbb{R}^{m_I \times n}$ represents the non-penetration condition on the contact zones. Notice that $\mathscr{K}$ is convex and closed.

Let $\mathbf{K}_e \in \mathbb{R}^{n \times n}$ be a block diagonal matrix consisting of the elastic stiffness matrices $\mathbf{K}_e^{i,p}$ defined on each subdomain $\Omega^{i,p}$, $i = 1,2$, $p = 1,\ldots,s_i$. Due to the presence of the Dirichlet boundary conditions on both subdomains and the Korn inequality, we can define the energy norm on $\mathscr{V}$:

$$\|\mathbf{v}\|_e := \sqrt{\mathbf{v}^T \mathbf{K}_e \mathbf{v}} = \sqrt{\sum_{i=1}^{2} \sum_{p=1}^{s_i} \mathbf{v}_{i,p}^T \mathbf{K}_e^{i,p} \mathbf{v}_{i,p}}, \quad \mathbf{v} = \left(\mathbf{v}_{1,1}^T, \ldots, \mathbf{v}_{1,s_1}^T, \mathbf{v}_{2,1}^T, \ldots, \mathbf{v}_{2,s_2}^T\right)^T \in \mathscr{V}.$$

Notice that the using of this norm is suitable from mechanical and mathematical points of view since some of the below estimates (mainly (6)) are independent of the domain decomposition and the discretization parameter h of the mesh.

The algebraic formulation of the contact elasto-plastic problem can be written as the following optimization problem [1]:

$$\text{Find } \mathbf{u} \in \mathscr{K} \ : \ J(\mathbf{u}) \leq J(\mathbf{v}) \quad \forall \mathbf{v} \in \mathscr{K}, \tag{3}$$

where

$$J(\mathbf{v}) := \Psi(\mathbf{v}) - \mathbf{f}^T \mathbf{v}, \quad \mathbf{v} \in \mathbb{R}^n. \tag{4}$$

Here the vector $\mathbf{f} = \left(\mathbf{f}_{1,1}^T, \ldots, \mathbf{f}_{1,s_1}^T, \mathbf{f}_{2,1}^T, \ldots, \mathbf{f}_{2,s_2}^T\right)^T \in \mathbb{R}^n$ represents the load consisting of the volume and surface forces, and the initial stress state. The functional Ψ represents the inner energy and has the structure

$$\Psi(\mathbf{v}) = \left(\Psi_{1,1}(\mathbf{v}_{1,1})^T, \ldots, \Psi_{1,s_1}(\mathbf{v}_{1,s_1})^T, \Psi_{2,1}(\mathbf{v}_{2,1})^T, \ldots, \Psi_{2,s_2}(\mathbf{v}_{2,s_2})^T\right)^T.$$

Further Ψ is a potential to the non-linear elasto-plastic operator $F : \mathbb{R}^n \to \mathbb{R}^n$, i.e. $D\Psi(\mathbf{v}) = F(\mathbf{v})$, $\forall \mathbf{v} \in \mathbb{R}^n$. The function F is generally nonsmooth but Lipschitz continuous. It enables us to define a generalized derivative $K : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ of F in the sense of Clark, i.e. $K(\mathbf{v}) \in \partial F(\mathbf{v})$, $\mathbf{v} \in \mathbb{R}^n$. Notice that $K(\mathbf{v})$ is symmetric, block diagonal and sparse matrix. Moreover the following properties of F and K hold [11]:

(i)
$$F(\mathbf{v}+\mathbf{w}) - F(\mathbf{v}) = \int_0^1 K(\mathbf{v}+\theta\mathbf{w})\mathbf{w} \, \mathrm{d}\theta \quad \forall \mathbf{v}, \mathbf{w} \in \mathbb{R}^n. \tag{5}$$

(ii) $K(\mathbf{v})$ is uniformly positive definite and bounded with respect to $\mathbf{v} \in \mathscr{V}$:

$$\exists \nu \in (0,1) : \quad \nu \|\mathbf{w}\|_e^2 \leq \mathbf{w}^T K(\mathbf{v}) \mathbf{w} \leq \|\mathbf{w}\|_e^2 \quad \forall \mathbf{v}, \mathbf{w} \in \mathscr{V}. \tag{6}$$

(iii) F is strongly semismooth [9] on $\mathscr{V}$, which yields that for any $\mathbf{v} \in \mathscr{V}$ and any of sufficiently small $\mathbf{w} \in \mathscr{V}$:

$$F(\mathbf{v}+\mathbf{w}) - F(\mathbf{v}) - K(\mathbf{v}+\mathbf{w})\mathbf{w} = O(\|\mathbf{w}\|_e^2). \tag{7}$$

Notice that (5) and (6) yield that Ψ is coercive and strictly convex on $\mathscr{V}$. Hence the problem (4) has a unique solution and can be equivalently written as the following variational inequality:

$$\text{Find } \mathbf{u} \in \mathscr{K} \ : \ F(\mathbf{u})^T (\mathbf{v} - \mathbf{u}) \geq \mathbf{f}^T (\mathbf{v} - \mathbf{u}) \quad \forall \mathbf{v} \in \mathscr{K}. \tag{8}$$

The estimate (7) will be important for showing that the semi-smooth Newton method defined in the next section has a local quadratic convergence.

3 Semi-smooth Newton method for optimization problem

The investigated problem (3) contains two nonlinearities – the non-quadratic functional J (due to Ψ) and the non-penetration conditions including in the convex set $\mathscr{K}$. By the semismooth Newton method, we will approximate Ψ by a quadratic functional similarly as in the Taylor expansion:

$$\Psi(\mathbf{u}) \approx \Psi(\mathbf{u}^k) + F(\mathbf{u}^k)^T(\mathbf{u}-\mathbf{u}^k) + \frac{1}{2}(\mathbf{u}-\mathbf{u}^k)^T K(\mathbf{u}^k)(\mathbf{u}-\mathbf{u}^k),$$

for a given approximation $\mathbf{u}^k \in \mathscr{K}$ of the solution $\mathbf{u}$ to the problem (3). Let us denote $\mathbf{f}_k = \mathbf{f} - F(\mathbf{u}^k)$, $\mathbf{K}_k = K(\mathbf{u}^k)$ and define:

$$\mathscr{K}_k := \mathscr{K} - \mathbf{u}^k = \left\{ \mathbf{v} \in \mathbb{R}^n \,;\, \mathbf{B}_E\mathbf{v} = \mathbf{o},\ \mathbf{B}_I\mathbf{v} \le \mathbf{c}_{I,k},\ \mathbf{c}_{I,k} := \mathbf{c}_I - \mathbf{B}_I\mathbf{u}^k \right\},$$

$$J_k(\mathbf{v}) := \frac{1}{2}\mathbf{v}^T\mathbf{K}_k\mathbf{v} - \mathbf{f}_k^T\mathbf{v}, \quad \mathbf{v} \in \mathscr{K}_k. \tag{9}$$

Then the Newton step is following:

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \delta\mathbf{u}^k, \quad \mathbf{u}^{k+1} \in \mathscr{K},$$

where $\delta\mathbf{u}^k \in \mathscr{K}_k$ is a unique minimum of J_k on $\mathscr{K}_k$:

$$J_k(\delta\mathbf{u}^k) \le J_k(\mathbf{v}) \qquad \forall \mathbf{v} \in \mathscr{K}_k, \tag{10}$$

or equivalently $\delta\mathbf{u}^k \in \mathscr{K}_k$ solves the following inequality:

$$\left(\mathbf{K}_k\delta\mathbf{u}^k\right)^T(\mathbf{v}-\delta\mathbf{u}^k) \ge \mathbf{f}_k^T(\mathbf{v}-\delta\mathbf{u}^k) \quad \forall \mathbf{v} \in \mathscr{K}_k. \tag{11}$$

Notice that if we substitute $\mathbf{v} = \mathbf{u}^{k+1} \in \mathscr{K}$ into (8) and $\mathbf{v} = \mathbf{u} - \mathbf{u}^k \in \mathscr{K}_k$ into (11), then by adding we obtain the inequality

$$\left(K(\mathbf{u}^k)\delta\mathbf{u}^k\right)^T(\mathbf{u}-\mathbf{u}^{k+1}) \ge \left(F(\mathbf{u}) - F(\mathbf{u}^k)\right)^T(\mathbf{u}-\mathbf{u}^{k+1}),$$

which can be arranged into the form

$$(\mathbf{u}^{k+1}-\mathbf{u})^T K(\mathbf{u}^k)(\mathbf{u}^{k+1}-\mathbf{u}) \le \left(F(\mathbf{u}^k) - F(\mathbf{u}) - K(\mathbf{u}^k)(\mathbf{u}^k-\mathbf{u})\right)^T(\mathbf{u}-\mathbf{u}^{k+1}).$$

Hence one can simply derive local quadratic convergence of the semi-smooth Newton method by (6) and (7) provided that $\mathbf{u}^k$ is sufficiently close to $\mathbf{u}$.

4 TFETI method for the inner problem

Notice that the structures and properties of the matrices $\mathbf{K}_k \in \mathbb{R}^{n\times n}$, $k = 0,1,2,\ldots$, are very similar to the corresponding elastic matrix $\mathbf{K}_e$ as follows from Section 2. Therefore we can solve the inner problem (10) in the same way as a contact problem with elastic bodies, see e.g. [4, 5].

Here we use the TFETI domain decomposition method for solving (10). For more detail see e.g. [3] and [1]. The method is based on enforcing all the constraints by the Lagrange multipliers. In particular, we use two types of Lagrange multipliers, namely $\lambda_I \in \mathbb{R}^{m_I}$, $\lambda_I \geq \mathbf{o}$ related to the non-penetration condition, $\lambda_E \in \mathbb{R}^{m_E}$ related to the "gluing" and Dirichlet conditions. To simplify the notation, we denote

$$\lambda = \begin{bmatrix} \lambda_E \\ \lambda_I \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_E \\ \mathbf{B}_I \end{bmatrix}, \quad \mathbf{c}_k = \begin{bmatrix} \mathbf{o} \\ \mathbf{c}_{I,k} \end{bmatrix},$$

and

$$\Lambda = \{\lambda = (\lambda_E^T, \lambda_I^T)^T \in \mathbb{R}^{m_E+m_I} : \ \lambda_I \geq \mathbf{o}\}.$$

Then the Lagrangian associated with problem (10) reads as

$$L_k(\mathbf{v},\lambda) = \frac{1}{2}\mathbf{v}^T\mathbf{K}_k\mathbf{v} - \mathbf{f}_k^T\mathbf{v} + \lambda^T(\mathbf{B}\mathbf{v} - \mathbf{c}_k), \quad \mathbf{v} \in \mathbb{R}^n, \ \lambda \in \Lambda. \tag{12}$$

Using the convexity of the cost function and constraints, we can use the classical duality theory to reformulate problem (10) to get

$$J_k(\delta\mathbf{u}^k) = \min_{\mathbf{v}\in\mathscr{K}_k} J_k(\mathbf{v}) = \min_{\mathbf{v}\in\mathbb{R}^n} \sup_{\lambda\in\Lambda} L_k(\mathbf{v},\lambda) = \max_{\lambda\in\Lambda} \inf_{\mathbf{v}\in\mathbb{R}^n} L_k(\mathbf{v},\lambda) = \max_{\lambda\in\Lambda}\{-\Theta_k(\lambda)\}, \tag{13}$$

with

$$\Theta_k(\lambda) = \begin{cases} \frac{1}{2}\lambda^T\mathbf{B}\mathbf{K}_k^\dagger\mathbf{B}^T\lambda - \lambda^T(\mathbf{B}\mathbf{K}_k^\dagger\mathbf{f}_k - \mathbf{c}_k), & \mathbf{R}_k^T(\mathbf{f}_k - \mathbf{B}^T\lambda) = \mathbf{o}, \\ +\infty, & \text{otherwise}, \end{cases}$$

where $\mathbf{K}_k^\dagger$ is a pseudoinverse matrix to $\mathbf{K}_k$ and $\mathbf{R}_k \in \mathbb{R}^{n\times l}$ represents the null space of $\mathbf{K}_k$. More details to implementation of $\mathbf{B}\mathbf{K}_k^\dagger\mathbf{B}^T$ can be found in [6]. Thus the corresponding dual problem has the form:

$$\text{find } \lambda^k \in \Lambda: \quad \Theta_k(\lambda^k) \leq \Theta_k(\lambda) \quad \forall\lambda \in \Lambda. \tag{14}$$

We solve the dual problem by algorithm SMALSE-M [3]. The algorithm is based on active set strategy and it combines three steps: CG with preconditioning based on orthogonal projectors, expansion, and proportioning.

Once the solution λ^k of (14) is known, the solution of (10) can be evaluated in this way:

$$\delta\mathbf{u}^k = \mathbf{K}_k^\dagger(\mathbf{f} - \mathbf{B}^T\lambda^k) + \mathbf{R}_k\alpha_k, \quad \alpha_k = (\mathbf{R}_k^T\overline{\mathbf{B}}^T\overline{\mathbf{B}}\mathbf{R}_k)^{-1}\mathbf{R}_k^T\overline{\mathbf{B}}^T(\overline{\mathbf{c}}_k - \overline{\mathbf{B}}\mathbf{K}_k^\dagger(\mathbf{f}_k - \mathbf{B}^T\lambda^k)),$$

where the matrix $\overline{\mathbf{B}}$ and the vector $\bar{\mathbf{c}}_k$ are formed by the rows of $\mathbf{B}$ and $\mathbf{c}_k$ corresponding to all equality constraints and all active inequality constraints.

Notice that we use in fact the inexact Newton method with respect to computing of $\delta\mathbf{u}^k$.

5 Numerical experiments

In this section we illustrate the strong parallel scalability and the performance of numerical scalability of our approach on a numerical example. The geometry of the problem is depicted in Figure 1. The sizes of the bodies are $3000 \times 1000 \times 1000$. We use regular meshes generated in MatSol [8]. The Young modulus, the Poisson ratio, the initial yield stress for the von Mises criterion, and the hardening modulus are $E^i = 210\,000$, $\nu^i = 0.29$, $\sigma_y^i = 450$, and $H_m^i = 10\,000$, $i = 1,2$, respectively. The indicated traction force prescribed in the vertical direction is $g(x) = 150$, $x \in \Gamma_N^2$. The initial stress (or plastic strain) state is equal to zero.

The proposed algorithms were parallelized using Matlab Distributed Computing Server and Matlab Parallel Toolbox. For all computations we use 28 cores with 2GB memory per core of the HP Blade system, model BLc7000. The stopping criterion of the Newton method is $\frac{\|\mathbf{u}^{k+1}-\mathbf{u}^k\|_e}{\|\mathbf{u}^{k+1}\|_e+\|\mathbf{u}^k\|_e} < 10^{-4}$ (see e.g. [7] or [11]). The stopping criterion for the SMALSE-M algorithm is described in [3]. We use the tolerance 10^{-7} for SMALSE-M.

The strong parallel scalability is depicted in Table 1. Here we consider the mesh with 174 902 nodes and 162 000 hexahedrons. The bodies are decomposed into 162 subdomains by MatSol. The number of primal variables is 646 866 and the number of dual variables is 130 189.

Number of cores	3	7	14	28
Number of plastic elems.	151 300	151 300	151 300	151 300
Number of Newton iters.	6	6	6	6
Total number of SMALSE-M iters.	67	67	67	67
Total number of multi. by Hessian	3 726	3 726	3 726	3 726
Time for last Newton iter.	6 976	1 259	778	537
Total time [sec]	26 828	6 481	4 091	2 926

Table 1 Strong paralel scalability.

In Table 2 we report "the numerical scalability" for different mesh levels. The most important is row with total number of multiplication by Hessian, where we can see, that the number of iterations grows only moderately. The total times are not mutually comparable since we could not keep a constant number of subdomain per one core due to the limitation on maximal number of the core.

Distribution of the von Mises stress and the total displacement for the finest mesh are depicted in Figures 2 and 3.

Mesh level	1	2	3	4
Mesh nodes	7 502	53 802	174 902	406 802
Mesh elements	6 000	48 000	162 000	384 000
Number of subdomains	6	48	162	384
Number of cores	4	25	28	28
Primal variables	23 958	191 664	646 866	1 533 312
Dual variables	2 453	33 933	130 189	326 969
Number of plastic elems.	6 624	48 141	151 300	356 384
Number of Newton iters.	6	6	6	6
Total number of SMALSE-M iters.	153	88	67	67
Total number of multi. by Hessian	1 951	3 106	3 726	5 375
Time for last Newton iter.	41	141	537	1 758
Total time [sec]	287	683	2 926	9 318

Table 2 Performance of ”the numerical scalability”.

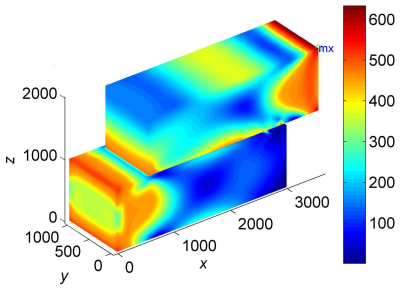


Fig. 2 von Mises stress distribution

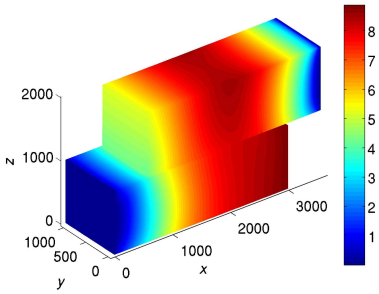


Fig. 3 total displacement

6 Conclusion

In this paper, we proposed a numerical method for solving contact elasto-plastic problems based on TFETI method and demonstrate its parallel and numerical scalability on a numerical example. The numerical realization and implementation of the problem were newly included into the MatSol library. In fact, the proposed method can be used or can be as a part of other contact inelastic problems than the considered frictionless contact problem of von Mises' elasto-plastic bodies with isotropic hardening.

Acknowledgements This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the project SPOMECH - Creating a multidisciplinary R&D team for reliable solution of mechanical problems, reg. no. CZ.1.07/2.3.00/20.0070 within Operational Programme 'Education for competitiveness' funded by Structural Funds of the European Union and state budget of the Czech Republic.

References

1. Cermak, M.: Scalable algorithms for solving elasto-plastic problems. Ph.D. thesis, VSB-TU Ostrava (2012)
2. Dostál, Z., Horák, D., Kučera, R.: Total feti - an easier implementable variant of the feti method for numerical solution of elliptic pde. Communications in Numerical Methods in Engineering **22**, 1155–1162 (2006)
3. Dostál, Z., Kozubek, T.: An optimal algorithm and superrelaxation for minimization of a quadratic function subject to separable convex constraints with applications. Mathematical Programming **135**, 195–220 (2012)
4. Dostál, Z., Kozubek, T., Markopoulos, A., Brzobohatý, T., Vondrák, V., Horyl, P.: Theoretically supported scalable tfeti algorithm for the solution of multibody 3d contact problems with friction. CMAME **205-208**, 110–120 (2012)
5. Dostál, Z., Kozubek, T., Vondrák, V., Brzobohatý, T., Markopoulos, A.: Scalable tfeti algorithm for the solution of multibody contact problems of elasticity. International Journal for Numerical Methods in Engineerign **82**, 1384–1405 (2012)
6. Farhat, C., Mandel, J., Roux, F.X.: Optimal convergence properties of the feti domain decomposition method. Comput. Methods Appl. Mech. Eng. **115**, 365–385 (1994)
7. Gruber, P.G., Valdman, J.: Solution of one-time step problems in elastoplasticity by a slant newton method. SIAM J. Sci. Comp. **31**, 1558–1580 (2009)
8. Kozubek, T., Markopoulos, A., Brzobohatý, T., Kučera, R., Vondrák, V., Dostál, Z.: Matsol - matlab efficient solvers for problems in engineering. Available from http://matsol.vsb.cz/ (2012)
9. Qi, L., Sun, J.: A nonsmooth version of newton's method. Mathematical Programming **58**, 353–367 (1993)
10. de Souza Neto, E.A., Perić, D., Owen, D.R.J.: Computational methods for plasticity, Theory and applications. John Wiley, West Sussex (2008)
11. Sysala, S.: Application of a modified semismooth newton method to some elasto-plastic problems. Mathematic Computation and Simulation **82**, 2004–2021 (2012)
12. Wohlmuth, B.: Variationally consistent discretization schemes and numerical algorithms for contact problems. Acta Numerica **20**, 569–734 (2011)

Nonlinear Transmission Conditions for time Domain Decomposition Method

P. Linel[1] and D. Tromeur-Dervout[2]

1 Introduction

We developed parallel time domain decomposition methods to solve systems of linear ordinary differential equations (ODEs) based on the Aitken-Schwarz [7] or primal Schur complement domain decomposition methods [6]. The methods claim the transformation of the initial value problem in time defined on $]0,T]$ into a time boundary values problem. Let $f(t,y(t))$ be a function belonging to $\mathscr{C}^1(\mathbb{R}^+,\mathbb{R}^d)$ and consider the Cauchy problem for the first order ODE:

$$\left\{\dot{y} = f(t,y(t)), t \in]0,T], \; y(0) = y_0 \in \mathbb{R}^d. \right. \tag{1}$$

The time interval $[0,T]$ is split into p time slices $S_i = [T_{i-1}^+, T_i^-]$, with $T_0^+ = 0$ and $T_p^- = T^-$. The difficulty is to match the solutions $y_i(t)$ defined on S_i at the boundaries T_{i-1}^+ and T_i^-. Most of time domain decomposition methods are shooting methods [1] where the jumps $y_i(T_i^-) - y_{i+1}(T_i^+)$ are corrected by a sequential process which is propagated in the forward direction (i.e. the correction on the time slice S_{i-1} is needed to compute the correction on time slice S_i). Our approach consists in breaking the sequentiality of the solution's initial value updating for each time slice. For this, we transform the initial value problem (IVP) into a boundary values problem (BVP) leading to a second order ODE:

$$\ddot{y}(t) = g(t,y(t)) \stackrel{def}{=} \frac{\partial f}{\partial t}(t,y) + f(t,y(t))\frac{\partial f}{\partial y}(t,y(t)), t \in]0,T[, \tag{2a}$$

$$y(0) = y_0, \tag{2b}$$

$$y(T) = \beta. \tag{2c}$$

Nevertheless, the difficulty in solving equation (2) is that β is not given by the original IVP. To overcome the lack of knowledge of β, we proposed to set this value by using an iterative Schwarz domain decomposition method with no overlapping. For sake of simplicity, let us consider only one domain S_1. Given a,b in $\mathbb{R}^+$ with $a < b$, we denote $\overline{[a,b]}$ to indicate that the time interval must be traveled in the back-

[1] University of Rochester Medical Center,Dept of Biostatistics and Computational Biology, Saunders Research Building, 265 Crittenden Blvd. Rochester, NY 14642, USA, e-mail: Patrice_Linel@URMC.Rochester.edu [2] University of Lyon, University Lyon 1, CNRS, UMR5208 Institut Camille Jordan, 43 Bd du 11 Novembre 1918, 69622 Villeurbanne cedex, France, e-mail: Damien.tromeur-dervout@univ-lyon1.fr

ward direction. We first symmetrize the time interval S_1 providing $\bar{S_1} = \overline{[0^+, T^-]}$. A symmetric time integration scheme, like the second order implicit Störmer-Verlet symmetric scheme, is then required to perform a backward time integration onto the symmetrized interval to come back to the initial state. Then classical domain decomposition methods can be applied such the multiplicative Schwarz method with no overlapping time slices with Dirichlet-Neumann (associated to the Laplacian in time) transmission conditions (T.C.) for linear system of ODE (or PDE [8]). As proved in [7] the convergence/divergence of the error at the boundaries of this Schwarz time DDM can be accelerated by the Aitken technique to the right solution when $f(t,y(t))$ is linear.

This paper treats the case where $f(t,y(t))$ is nonlinear. Then the multiplicative Schwarz algorithm generates at the boundary of time slices a nonlinear vectorial sequence. We replaced in [5] the Aitken's acceleration of the convergence by the ε-topological algorithm [3] that has been designed to extrapolate the convergence of such nonlinear sequences. Some enhancement of the convergence have been obtained but the number of Schwarz iterations is still too large to obtain an efficient method. This leads us to think again about the transmission conditions between time slices. When systems of nonlinear ODEs are under consideration, we show in the next section that the Dirichlet-Neumann T.C. (associated to the time Laplacian operator only) at boundary time slices are not the right choice. The Neumann boundary condition has to be replaced by a nonlinear boundary condition preserving an invariant of the solution. These nonlinear T.C. differ from the optimized nonlinear T.C. present in the waveform relaxation of [4]. In section 3, we show the pure linear behavior of the multiplicative Schwarz with a combination of the nonlinear T.C. and the Dirichlet condition by demonstrating that the operator associated to the error does not depend of the iteration. This operator links the transmission conditions of all the time slices, allowing to solve the problem on all time slices in the same time using the Aitken acceleration of the convergence. Some perspectives of this work are given in the conclusion.

2 What are the right T.C. in the nonlinear case?

Let us first give a new formulation of the equation (2) assuming that $f(t,y(t)$ is scalar and $f^{-1}(t,y(t))$ exists. Then one can consider the problem:

$$-\frac{d}{dt}[-f^{-1}(t,y(t))\frac{d}{dt}y(t)] = -\frac{d}{dt}(-1) = 0, t \in]0,T[, y(0) = 0, \tag{3a}$$

$$y(T) = 1. \tag{3b}$$

where we imposed a Dirichlet B.C. at the time $t = T$ for the sake of simplicity. Then the multiplicative Schwarz with Neumann (associated to the Laplacian operator)-Dirichlet T.C. applied to $[0,T] = [0,1] = [0,\Gamma] \cup [\Gamma,1]$ with $\Gamma = 3/5$ writes:

$$-\frac{d}{dt}[-f^{-1}(t,y_1^{n+\frac{1}{2}}(t))\frac{d}{dt}y_1^{n+\frac{1}{2}}(t)]=0,\ t\in]0,\Gamma[,y_1^{n+\frac{1}{2}}(0)=0, \tag{4a}$$

$$y_1^{n+1}(\Gamma)=\alpha^n=y_2^n(\Gamma), \tag{4b}$$

and

$$-\frac{d}{dt}[-f^{-1}(t,y_2^{n+1}(t))\frac{d}{dt}y_2^{n+1}(t)]=0,\ t\in]\Gamma,1[,y_2^{n+1}(1)=1, \tag{5a}$$

$$\frac{d}{dt}y_2^{n+1}(\Gamma)=\beta^{n+1}=\frac{d}{dt}y_1^{n+\frac{1}{2}}(\Gamma). \tag{5b}$$

Let us consider $f(t,y(t))=\sqrt{y(t)}$ then the exact solution is $y(t)=t^2$ and $y(3/5)=\overline{\alpha}=9/25$. The exact solution of the Neumann-Dirichlet writes:

$$y_1^{n+\frac{1}{2}}(t)=\frac{25}{9}t^2\alpha^n\rightarrow\frac{d}{dt}y_1^{n+1}(\frac{3}{5})=\frac{10}{3}\alpha^n. \tag{6}$$

$$y_2^{n+1}(t)=\begin{cases}\frac{25}{4}r_1^2t^2+\frac{5}{2}r_1t(-5r_1-2)+\frac{1}{4}(-5r_1-2)^2,\\ \frac{25}{4}r_2^2t^2+\frac{5}{2}r_2t(-5r_2+2)+\frac{1}{4}(-5r_2+2)^2.\end{cases} \tag{7}$$

where r_1 (respectively r_2) is the root of $3r_1^2+3r_1+2\alpha=0$ (respectively $3r_2^2-r_2+2\alpha=0$). The sequence (α^n) satisfies one of the equation that follows:

$$\alpha^{n+1}=\begin{cases}f_1(\alpha^n)=1/2-(1/6)\sqrt{9-24\alpha^n}-(2/3)\alpha^n,\\ f_2(\alpha^n)=1/2+(1/6)\sqrt{9-24\alpha^n}-(2/3)\alpha^n.\end{cases} \tag{8}$$

If $\alpha^{n+1}=f_1(\alpha^n)$ then the sequence converges toward the fixed point $\overline{\alpha}_1=f_1(\overline{\alpha}_1)=0$ as $|f_1^{(1)}(\overline{\alpha}_1)|<1$. But $\overline{\alpha}_1\neq\overline{\alpha}$. If $\alpha^{n+1}=f_2(\alpha^n)$ then $\overline{\alpha}_2=f_2(\overline{\alpha}_2)=\overline{\alpha}$, but $|f_2^{(1)}(\overline{\alpha}_2)|>1$ and the function is not contractive. In both cases the multiplicative Schwarz will not converge with these transmission conditions.

If we replace Equation (5b) by Equation (9b):

$$-\frac{d}{dt}[-f^{-1}(t,y_2^{n+1}(t))\frac{d}{dt}y_2^{n+1}(t)]=0,\ t\in]\Gamma,1[,y_2^{n+1}(1)=1, \tag{9a}$$

$$f^{-1}(\Gamma,y_2^{n+1}(\Gamma))\frac{d}{dt}y_2^{n+1}(\Gamma)=\beta^{n+1}=f^{-1}(\Gamma,y_1^{n+\frac{1}{2}}(\Gamma))\frac{d}{dt}y_1^{n+\frac{1}{2}}(\Gamma). \tag{9b}$$

The sequence (α^n) of the Dirichlet condition satisfies :

$$\alpha^{n+1}=\begin{cases}0, & \alpha^n>\frac{9}{4},\\ \frac{4}{9}\alpha^n-\frac{4}{3}\sqrt{\alpha^n}+1, & 0\leq\alpha^n<\frac{9}{4}.\end{cases}\quad\text{,thus }\ \alpha^n\rightarrow\overline{\alpha}=\frac{9}{25}.$$

This result shows that we can not simplify the T.C. by only taking the matching of the time derivatives between time slices, even if the nonlinear function $f^{-1}(t,y(t))$ is continuous.

Coming back to the original formulation of the Schwarz algorithm for the second order ODE Equation (2), the T.C. to replace the transmission condition $\frac{d}{dt}y_1^{m+\frac{1}{2}}(T^-) = \frac{d}{dt}\overline{y}_1^m(T^-)$ should be the flux or co-normal derivative $f^{-1}(y_1^{m+\frac{1}{2}}(T^-))\frac{d}{dt}y_1^{m+\frac{1}{2}}(T^-) = -f^{-1}(\overline{y}_1^m(T^-))\frac{d}{dt}\overline{y}_1^m(T^-)$, if $f^{-1}(y_1^{n+1}(T^-)) \neq 0$, else $\frac{d}{dt}y_1^{m+\frac{1}{2}}(T^-) = 0$. Moreover, this invariant of the problem, allows us to simplify the methodology too. We can impose (with assuming $f^{-1}(T^-,y(T^-)) \neq 0$) the B.C. $f^{-1}(T^-,y(T^-))\frac{d}{dt}y_(T^-) = 1$. Consequently, we do not need to symmetrize the time interval and then saving by a factor 2 the computational resources needed.

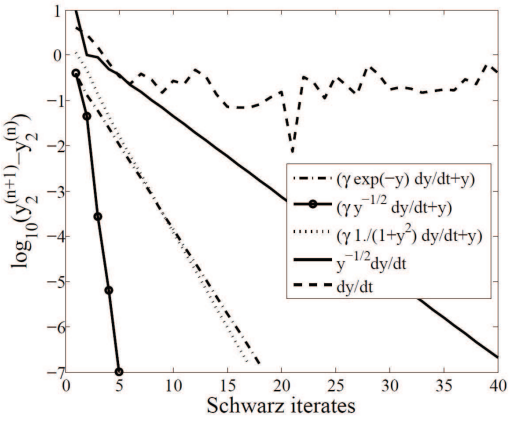


Fig. 1 Convergence/Divergence of the multiplicative Schwarz with respect to the T.C. $f^{-1}(t,y(t))\frac{d}{dt}y(t)$ with $f^{-1}(t,y(t)) = \{(\sqrt{y(t)})^{-1}, exp(-y(t)), \frac{1}{1+y^2(t)}\}$, or $\frac{d}{dt}y(t)$.

Figure 1 represents the numerical convergence of multiplicative Schwarz with the discretized nonlinear T.C. for the discretizing scheme associated to the Equation (3) with $f^{-1}(t,y(t)) = \{(\sqrt{y(t)})^{-1}, exp(-y(t)), \frac{1}{1+y^2(t)}\}$. It exhibits that the convergence behavior is purely linear for this problem with two time slices and one artificial interface. The T.C. with imposing the matching of $\frac{dy}{dt}(t)$ only does not converge as expected by the theory. The combining of the Dirichlet and relaxed flux for T.C. converges faster. We show in section 3 the pure linear behavior for the convergence of the multiplicative Schwarz for the time decomposition with this kind of nonlinear T.C. .

3 Pure linear convergence of the Time Schwarz DDM with nonlinear flux transmission conditions

Let us consider the problem Equation (3a) with Dirichlet B.C. at $t = 0$ and the invariant flux B.C. equal to 1 at $t = T$. Then we split the time interval $[0,T[$ into p time slices of size $H = T/p$ and we apply the multiplicative Schwarz algorithm with Dirichlet B.C. at $t = T_{i-1}^+$ and a combination of a Dirichlet and the invariant flux B.C. at $t = T_i^-$ times a parameter γ:

$$\frac{d}{dt} f^{-1}(t, y_i^{n+\frac{1}{2}}(t)) \frac{d}{dt} y_i^{n+\frac{1}{2}}(t) = 0, t \in S_i, \tag{10a}$$

$$y_i^{n+\frac{1}{2}}(T_{i-1}^+) = y_{i-1}^n(T_{i-1}^-), \tag{10b}$$

$$y_i^{n+\frac{1}{2}}(T_i^-) + \gamma f^{-1}(T_i^-, y_i^{n+\frac{1}{2}}(t)) \frac{d}{dt} y_i^{n+\frac{1}{2}}(T_i^-) = y_{i+1}^n(T_i^+) + \gamma f^{-1}(T_i^+, y_{i+1}^n(T_i^+)) \frac{d}{dt} y_{i+1}^n(T_i^+). \tag{10c}$$

Following the idea of [2], we use the Kirchoff transformation by introducing new variables $u_i(t)$ such that

$$u_i(t) := \Theta(y_i(t)) = \int^{y_i(t)} f^{-1}(t, z(t)) dz \text{ a.e. in } S_i. \tag{11}$$

Then $f^{-1}(t, y_i(t)) \frac{d}{dt} y_i(t) = \frac{d}{dt} u_i(t)$. Here the $f^{-1}(t, z(t))$ is taken sufficiently continuous such that the value of $\Theta(y(t^-)) = \Theta(y(t^+))$ and an equality on "y" traduces an equality on "u". Schwarz Algorithm (10) can be rewritten as:

$$\frac{d^2}{dt^2} u_i^{n+\frac{1}{2}}(t) = 0, t \in S_i, \tag{12a}$$

$$u_i^{n+\frac{1}{2}}(T_{i-1}^+) = \eta_i^n \stackrel{def}{=} u_{i-1}^n(T_{i-1}^-), \tag{12b}$$

$$u_i^{n+\frac{1}{2}}(T_i^-) + \gamma \frac{du_i^{n+\frac{1}{2}}}{dt}(T_i^-) = \chi_i^n \stackrel{def}{=} u_{i+1}^n(T_i^+) + \gamma \frac{du_{i+1}^n}{dt}(T_i^+). \tag{12c}$$

We can show that the B.C. of this multiplicative Schwarz converge purely linearly to the B.C. associated to the solution. The error $e_i = u_i - u$ satisfies

$$\frac{d^2}{dt^2} e_i^{n+\frac{1}{2}}(t) = 0, t \in S_i, \tag{13a}$$

$$e_i^{n+\frac{1}{2}}(T_{i-1}^+) = e_{i-1}^n(T_{i-1}^-) = \alpha_i^n \stackrel{def}{=} \eta_i^n - \eta_i^\infty, \tag{13b}$$

$$e_i^{n+\frac{1}{2}}(T_i^-) + \gamma \frac{de_i^{n+\frac{1}{2}}}{dt}(T_i^-) = e_{i+1}^n(T_i^+) + \gamma \frac{de_{i+1}^n}{dt}(T_i^+) = \beta_i^n \stackrel{def}{=} \chi_i^n - \chi_i^\infty. \tag{13c}$$

The error $e_i(t)$ writes $e_i(t) = a_i\, t + b_i$ with:

$$a_i = \frac{\beta_i^n - \alpha_i^n}{\gamma + H}, \text{ and } b_i = -\frac{(\beta_i^n - \alpha_i^n)}{\gamma + H}\, T_{i-1}^+ + \alpha_i^n. \tag{14}$$

For the sake of simplicity, let us take $p = 6$. We have $\alpha_1^n = 0$ and $\beta_6^n = 0$. Then one can write: $\Xi_1^{n+\frac{1}{2}} := (\beta_1^{n+\frac{1}{2}}, \alpha_3^{n+\frac{1}{2}}, \beta_3^{n+\frac{1}{2}}, \alpha_5^{n+\frac{1}{2}}, \beta_5^{n+\frac{1}{2}})^T = \mathbb{P}_1 \Xi_2^n$
and $\Xi_2^n := (\alpha_2^n, \beta_2^n, \alpha_4^n, \beta_4^n, \alpha_6^n)^T = \mathbb{P}_2 \Xi_1^{n-\frac{1}{2}}$ with:

$$\mathbb{P}_1 = \frac{1}{\gamma + H} \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ \gamma & H & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & \gamma & H & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \text{ and } \mathbb{P}_2 = \frac{1}{\gamma + H} \begin{pmatrix} H & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & \gamma & H & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & \gamma & H \end{pmatrix}. \tag{15}$$

The matrices $\mathbb{P}_1$ and $\mathbb{P}_2$ do not depend on the iteration n, and are invertible with an appropriate choice of γ. The matrix $\mathbb{P} = \mathbb{P}_1 \mathbb{P}_2$ links all the B.C. associated to the odd time slices as follows:

$$\mathbb{P} = \frac{1}{(\gamma + H)^2} \begin{pmatrix} -H & -1 & 1 & 0 & 0 \\ \gamma H & -H & H & 0 & 0 \\ 0 & -\gamma & -H & -1 & 1 \\ 0 & \gamma^2 & \gamma H & -H & H \\ 0 & 0 & 0 & -\gamma & -H \end{pmatrix}. \tag{16}$$

Consequently the multiplicative Schwarz algorithm converges or diverges purely linearly and the right B.C. associated with the solution can be extrapolated with the Aitken's acceleration of convergence technique using this convergence or divergence behavior. By setting $\Lambda_1^{n+\frac{1}{2}} \stackrel{def}{=} (\chi_1^{n+\frac{1}{2}}, \eta_3^{n+\frac{1}{2}}, \chi_3^{n+\frac{1}{2}}, \eta_5^{n+\frac{1}{2}}, \chi_5^{n+\frac{1}{2}})^T$, the Aitken's extrapolation, with the identity matrix $\mathbb{I}$, writes: $\Lambda_1^{\infty} = (\mathbb{I} - \mathbb{P})^{-1}(\Lambda_1^{\frac{3}{2}} - \mathbb{P}\Lambda_1^{\frac{1}{2}})$. For $H = 1$ and $\gamma = 0.5$ the eigenvalues of $\mathbb{P}$ are with 4 significant digits:
$\{-0.1413 \pm 0.2478\, i, -0.2608, -0.2221 \pm 0.1496\, i\}$ which shows the convergence of the multiplicative Schwarz.

Remark 1. We can not impose the flux T.C. only at the end of time slices because the flux B.C. at the last time slices then will impose $a_i = 0, \forall i$. Consequently we would have a sequential propagation of the right B.C. at each Schwarz iterate.

Remark 2. As we have $\frac{d}{dt} u_{i+1}^n(T_i^+) = 1$ then Equation (10c) can be replaced by:

$$u_i^{n+\frac{1}{2}}(T_i^-) + \gamma \frac{d}{dt} u_i^{n+\frac{1}{2}}(T_i^-) = \chi_i^n \stackrel{def}{=} u_{i+1}^n(T_i^+) + \gamma. \tag{17}$$

4 Numerical implementation and result

In order to implement the multiplicative Schwarz, we still use Equation (2a) with using the Störmer-Verlet second order in time implicit scheme. Considering $N+1$ regular time steps Δt on each time slice S_i, and $z_j \simeq y_i(T_{i-1}^+ + j\Delta t)$, the flux T.C. given by Equation (17) is discretized in time with the second order scheme with $f_N^{-1} = f(T_i^-, z_N)^{-1}$:

$$y_i(T_i^-) + \gamma f(T_i^-, y_i(T_i^-))^{-1} \frac{dy_i}{dt}(T_i^-) \simeq z_N + \gamma f_N^{-1}(\frac{3}{2}z_N - 2z_{N-1} + \frac{1}{2}z_{N-2}). \quad (18)$$

The local problem on each time slice consists in searching the zero of the function $F(z_0,\ldots,z_N) = 0$ including the two T.C. for $j = 0$ and $j = N$ with a Newton method with a stopping criterion set to be 10^{-7}. The Jacobian matrix of F is mainly a tridiagonal matrix when we applied a Gaussian elimination of the term in position $N, N-2$. Moreover the nonlinearity is concentrated in the scheme only on the diagonal of the Jacobian and on the last row. An initial solution is computed on a regular coarse time mesh with the Newton stopping criterion set to be 9.10^{-2}. Then the Kirshoff transformation Θ is applied to the T.C. Y^i (of odd time slices) in order to obtain the acceleration matrix $\mathbb{P}_\Theta$. Next, the Aitken acceleration is performed in the transformed space (associated to the Kirshoff transformation) and the accelerated T.C. Y^∞ on odd time slices are retrieved with applying Θ^{-1} as follows:

$$Y^\infty := \Theta^{-1}((\mathbb{I} - \mathbb{P}_\Theta)^{-1}(\Theta(Y^2) - \mathbb{P}_\Theta\Theta(Y^1))). \quad (19)$$

Remark 3. This formula generalizes to the nonlinear case the Aitken-SVD [9]. In this last case, $\Theta(Y) = \mathbb{U}Y$ is the linear change of variable where $\mathbb{U}$ comes from the singular value decomposition $\mathbb{U}\Sigma\mathbb{V}^T$ of the T.C. arizing in the Schwarz iterations.

5 Conclusion

We obtained new nonlinear transmission conditions for our time domain decomposition which consists to apply classical multiplicative Schwarz algorithm on non-overlapping time slices. These T.C. make the multiplicative Schwarz algorithm having a pure linear convergence that allows it to be extrapolated to the T.C. satisfied by the searched solution. The method is for the moment applied to scalar problem, some extension to system of non linear ODEs is under investigation by using the definition of the inverse of a vector used in the ε-algorithm.

Acknowledgements This work was supported by the French National Agency of Research through the project ANR-12-MONU-0012 H2MNO4. Second author was backed to the Région Rhône-Alpes. This work used the HPC resources of Center for the Development of Parallel Scientific Computing (CDCSP) of University Lyon 1.

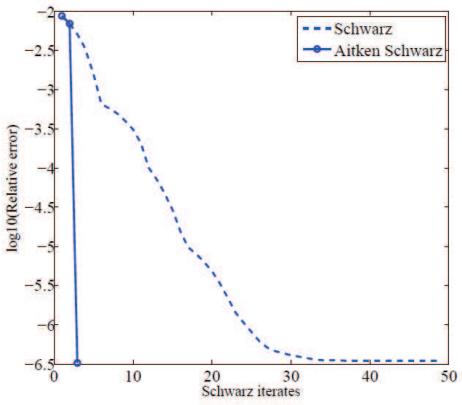


Fig. 2 Maximum of relative error between the Schwarz Dirichlet B.C. of odd time slices with the exact solution (dash line) and its acceleration by Aitken technique (solid line), with respect to the Schwarz iterations for $f(t,y(t)) = exp(y(t))$. Number of time slices is $p = 12$, $N = 81$, $\gamma = 20$.

References

1. Bellen, A., Zennaro, M.: Parallel algorithms for initial value problems for difference and differential equations. J. Comput. Appl. Math. **25**(3), 341–350 (1989). DOI 10.1016/0377-0427(89)90037-X
2. Berninger, H., Kornhuber, R., Sander, O.: Convergence behaviour of Dirichlet-Neumann and Robin methods for a nonlinear transmission problem. In: Domain decomposition methods in science and engineering XIX, *Lect. Notes Comput. Sci. Eng.*, vol. 78, pp. 87–98. Springer, Heidelberg (2011). DOI 10.1007/978-3-642-11304-8_8
3. Brezinski, C.: Convergence acceleration during the 20th century. J. Comput. Appl. Math. **122**(1-2), 1–21 (2000). DOI 10.1016/S0377-0427(00)00360-5. Numerical analysis 2000, Vol. II: Interpolation and extrapolation
4. Caetano, F., Gander, M.J., Halpern, L., Szeftel, J.: Schwarz waveform relaxation algorithms with nonlinear transmission conditions for reaction-diffusion equations. In: Domain decomposition methods in science and engineering XIX, *Lect. Notes Comput. Sci. Eng.*, vol. 78, pp. 245–252. Springer, Heidelberg (2011). DOI 10.1007/978-3-642-11304-8_27
5. Linel, P.: Méthodes de décomposition de domaines en temps et en espace pour la résolution de systèmes d' EDO non-linéaires. Ph.D. thesis, Universté Lyon 1 (2010)
6. Linel, P., Tromeur-Dervout, D.: Aitken-schwarz and schur complement methods for time domain decomposition. In: B. Chapman, F. Desprez, G. Joubert, A. Lichnewsky, F. Peters, T. Priol (eds.) Parallel Computing: From Multicores and GPU's to Petascale, *Advances in Parallel Computing*, vol. 19, pp. 75–82. IOS Press (2010)
7. Linel, P., Tromeur-Dervout, D.: Une méthode de décomposition en temps avec des schémas d'intégration réversible pour la résolution de systèmes d'équations différentielles ordinaires. C. R. Math. Acad. Sci. Paris **349**(15-16), 911–914 (2011). DOI 10.1016/j.crma.2011.07.002
8. Linel, P., Tromeur-Dervout, D.: Analysis of the time-schwarz ddm on the heat pde. Computers & Fluids **80**, 94–101 (2013). DOI 10.1016/j.compfluid.2012.04.023. URL `http://www.sciencedirect.com/science/article/pii/S0045793012001624`
9. Tromeur-Dervout, D.: Meshfree Adaptative Aitken-Schwarz Domain Decomposition with application to Darcy Flow. In: Topping, BHV and Ivanyi, P (ed.) Parallel, Distributed and Grid Computing for Engineering, *Computational Science Engineering and Technology Series*, vol. 21, pp. 217–250 (2009). DOI {10.4203/csets.21.11}