Finite Element Methods with patches and Applications

R. Glowinski[1], J. He[1], A. Lozinski[2], M. Picasso[2], J. Rappaz[2], V. Rezzonico[2], and J. Wagner[2]

[1] Department of Mathematics, University of Houston, USA
[2] Institute of Analysis and Scientific Computing, Ecole Polytechnique Fédérale de Lausanne, Switzerland
Correspondence to: J. Wagner, joel.wagner@epfl.ch

Summary. We present a new method [7] for numerically solving elliptic problems with multi-scale data using multiple levels of not necessarily nested grids. We use a relaxed iterative method which consists in calculating successive corrections to the solution in patches of finite elements. We analyse the spectral properties of the iteration operator [6]. We show how to evaluate the best relaxation parameter and what is the influence of patches size on the convergence of the method. Several numerical results in 2D and 3D are presented.

1 Introduction

In numerical approximation of elliptic problems by finite element method, a great precision of solutions is often required in certain regions of the domain in which the solution is defined. Efficient approaches are for instance adaptive mesh refinement techniques or domain decomposition methods. The objective of this paper is to present a method to solve numerically elliptic problems with multi-scale data using two levels of not necessarily nested grids.

Consider a multi-scale problem with sharp data in small sub-domains. We solve the problem on a coarse mesh of the computational domain. Therein we consider a patch (or multiple patches) with corresponding fine mesh wherein we would like to obtain more accuracy. Thus we calculate successively corrections to the solution in the patch. The coarse and fine discretizations are not necessarily conforming. The method is a domain decomposition method with complete overlapping. It resembles the Fast Adaptive Composite grid (FAC) method (see, e.g., [9]) or possibly a hierarchical method (see [3] for example). However it is of much more flexible use in comparison to the latter: in fact the discretizations do not need to be nested, conforming or structured. The idea of the method is strongly related to the Chimera method [4].

The outline of this paper is as follows. In Section 2 we introduce the algorithm and present an *a priori* estimate for the approximation (Prop. 1).

In Section 3 we present the convergence result for the method (Prop. 3) and give sharp results for the spectral properties of the iteration operator. We give a method to estimate the optimal relaxation parameter. In Section 4 we consider computational issues and discuss the implementation. Finally, in Section 5 we assess the efficiency of the algorithm in simple two-dimensional situations and give an illustration in 3D. The reader should note that this paper contains no proofs which can be found in [6].

2 Two-step algorithm

Let $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, be an open polygonal or polyhedral domain and consider a bilinear, symmetric, continuous and coercive form $a : H_0^1(\Omega) \times H_0^1(\Omega) \to \mathbb{R}$. The usual $H^1(\Omega)$-norm is equivalent to the a-norm defined by $||v|| = a(v,v)^{\frac{1}{2}}$, $\forall v \in H_0^1(\Omega)$. If $f \in H^{-1}(\Omega)$, due to Riesz'representation Theorem there exists a unique $u \in H_0^1(\Omega)$ such that

$$a(u,\varphi) = \langle f|\varphi\rangle, \quad \forall \varphi \in H_0^1(\Omega), \tag{1}$$

where $\langle\cdot|\cdot\rangle$ denotes the duality $H^{-1}(\Omega) - H_0^1(\Omega)$. Let us point out that (1) is the weak formulation of a problem of type $\mathcal{L}(u) = f$ in Ω, $u = 0$ on the boundary $\partial\Omega$ of Ω, where $\mathcal{L}(\cdot)$ is a second order, linear, symmetric, strongly elliptic operator.

A Galerkin approximation consists in building a finite dimensional subspace $V_{Hh} \subset H_0^1(\Omega)$, and solving the problem: Find $u_{Hh} \in V_{Hh}$ satisfying

$$a(u_{Hh},\varphi) = \langle f|\varphi\rangle, \quad \forall \varphi \in V_{Hh}. \tag{2}$$

In the following the construction of the space V_{Hh} is presented. We introduce a regular triangulation $\mathcal{T}_H$ of $\overline{\Omega}$, a union of triangles K of diameter less than or equal to H. Consider now a multi-scale situation with a solution that is very sharp, i.e. varies rapidly, in a small polygonal or polyhedral subdomain Λ of Ω, but smooth, i.e. varies slowly, in $\Omega \setminus \Lambda$. This means that the solution can be well approximated on a coarse mesh in $\Omega \setminus \Lambda$ but needs a fine mesh in Λ. We would like to stress that $\overline{\Lambda}$ is not necessarily the union of several triangles K of $\mathcal{T}_H$. Besides Λ can be determined in practice by an a priori knowledge of the solution behaviour or an a posteriori error estimator, for example. Let $\mathcal{T}_h$ be a regular triangulation of $\overline{\Lambda}$ with triangles K such that $\mathrm{diam}(K) \leq h$.

We define $V_H = \{\psi \in H_0^1(\Omega) : \psi|_K \in \mathbb{P}_r(K), \forall K \in \mathcal{T}_H\}$, and $V_h = \{\psi \in H_0^1(\Omega) : \psi|_K \in \mathbb{P}_s(K), \forall K \in \mathcal{T}_h$ and $\psi = 0$ in $\overline{\Omega} \setminus \Lambda\}$, where $\mathbb{P}_q(K)$ is the space of polynomials of degree $\leq q$ on triangle K. We set $V_{Hh} = V_H + V_h$. Let us observe that in practice, it is not possible to determine a finite element basis of V_{Hh}. The goal of our method is to evaluate efficiently u_{Hh} without having a basis of V_{Hh}, but only a basis of V_H and a basis of V_h.

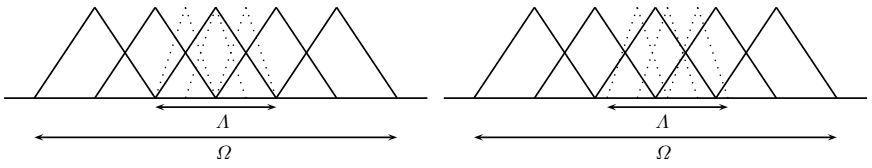
Before to show how to compute u_{Hh}, we give the following *a priori* estimate:

Proposition 1. *Let $q = \max(r, s) + 1$ and suppose that the solution u of (1) is in $H^q(\Omega)$. Then the approximate problem (2) has a unique solution u_{Hh} which satisfies the* a priori *error estimate*

$$||u - u_{Hh}|| \leq C \left(H^r ||u||_{H^q(\Omega \setminus \overline{\Lambda})} + h^s ||u||_{H^q(\Lambda)} \right), \tag{3}$$

where C is a constant independent of H, h and u.

Let us mention that a priori $V_H \cap V_h$ does not necessarily reduce to the element zero as shown in Fig. 1(a) where a 1D situation is illustrated by the hat functions in Ω and in Λ. In the case when $\mathcal{T}_H$ and $\mathcal{T}_h$ are not nested, as illustrated by Fig. 1(b) where we have translated the patch, it is not possible to easily exhibit a finite element-basis of V_{Hh} from the bases of V_H and V_h. Note also that moving from the situation depicted in Fig. 1(a) to the one in Fig. 1(b), the dimension of V_{Hh} increases by 1. All these difficulties impose an iterative method for solving problem (2).



(a) Nested elements. (b) Non-nested elements.

Fig. 1. Linear finite elements in 1D on Ω (plain lines) and Λ (dotted lines) .

So we suggest the following algorithm to compute u_{Hh}.

Algorithm 2

1. *Set $u^0 \in V_H$ such that $a(u^0, \varphi) = \langle f | \varphi \rangle, \quad \forall \varphi \in V_H$,*
   and choose $\omega \in (0; 2)$.
2. *For $n = 1, 2, 3, \ldots$ find*
   (i) $w_h \in V_h$ such that $a(w_h, \varphi) = \langle f | \varphi \rangle - a(u^{n-1}, \varphi), \quad \forall \varphi \in V_h$;
   $u^{n-\frac{1}{2}} = u^{n-1} + \omega w_h$;
   (ii)$w_H \in V_H$ such that $a(w_H, \varphi) = \langle f | \varphi \rangle - a(u^{n-\frac{1}{2}}, \varphi), \quad \forall \varphi \in V_H$;
   $u^n = u^{n-\frac{1}{2}} + \omega w_H$.

It is readily seen that this algorithm is a Schwarz type domain decomposition method [10] with complete overlapping but without any conformity between the meshes $\mathcal{T}_H$ and $\mathcal{T}_h$ (see for instance the work by Chan et al. [5]). It is similar to the Chimera or overset grid method [4, 11]. However, the algorithm presented in [4] is an additive method which can be changed to a multiplicative method equivalent to the above presented with $\omega = 1$.

Our multiplicative Schwarz method is also similar to a Gauss-Seidel method and can be put in the framework of the successive subspace correction algorithm by Xu and Zikatanov (see, e.g., [12]). The spaces V_H and V_h defined on the arbitrary triangulations $\mathcal{T}_H$ and $\mathcal{T}_h$ are not necessary orthognal nor share the only element zero as intersection. Note in particular that the sum which defines V_{Hh} is a priori not a direct sum. This property makes the above algorithm different from most known iterative schemes. For structured grid constellations, the algorithm resembles the FAC method (see for example the works from McCormick et al. [8]), or possibly a hierarchical method (see for example the papers from Yserentant [13], Bank et al. [2]) with a mortar method (see [1]).

We underline that the new aspect we introduce is to link the speed of convergence of this algorithm to the parameter $\tilde{\gamma}$, introduced here below, corresponding to the cosine of an abstract angle between the spaces V_h and V_H. Furthermore, an optimal relaxation keeps the method competitive in cases where the problem is badly conditioned (see Sect. 5).

3 Convergence analysis and consequences

We shall now analyse the convergence of the two-scale algorithm.[3] If $P_h : V_{Hh} \to V_h$ and $P_H : V_{Hh} \to V_H$ are orthogonal projectors from V_{Hh} upon V_h and V_H respectively with regard to the scalar product $a(\cdot,\cdot)$, and I denotes the identity operator in V_{Hh}, we set $B = (I - \omega P_H)(I - \omega P_h)$, and check that $u_{Hh} - u^n = B(u_{Hh} - u^{n-1})$.

We set $V_0 = V_H \cap V_h$ and $V_0^\perp$ the orthogonal complement of V_0 in V_{Hh} with respect to $a(\cdot,\cdot)$. We define $\tilde{V}_h = V_h \cap V_0^\perp$ and $\tilde{V}_H = V_H \cap V_0^\perp$. For $\omega \in (0;2)$ and $\tilde{\gamma} \in [0;1)$ defined by

$$\tilde{\gamma} = \begin{cases} \sup\limits_{\substack{v_h \in \tilde{V}_h, v_h \neq 0 \\ v_H \in \tilde{V}_H, v_H \neq 0}} \frac{a(v_h, v_H)}{||v_h|| ||v_H||}, & \text{if } V_h \neq V_0 \text{ and } V_H \neq V_0, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

we introduce the functions

$$\rho(\tilde{\gamma}, \omega) = \begin{cases} \frac{\omega^2 \tilde{\gamma}^2}{2} - \omega + 1 + \frac{\omega\tilde{\gamma}}{2}\sqrt{\omega^2\tilde{\gamma}^2 - 4\omega + 4}, & \text{if } \omega \leq \omega_0(\tilde{\gamma}), \\ \omega - 1, & \text{otherwise,} \end{cases} \tag{5}$$

where

$$\omega_0(\tilde{\gamma}) = \begin{cases} \frac{2 - 2\sqrt{1-\tilde{\gamma}^2}}{\tilde{\gamma}^2}, & \text{for } \tilde{\gamma} \in (0;1), \\ 1, & \text{for } \tilde{\gamma} = 0, \end{cases} \tag{6}$$

and $N(\tilde{\gamma}, \omega) = \omega(2-\omega)\tilde{\gamma}/2 + \sqrt{\omega^2(2-\omega)^2\tilde{\gamma}^2/4 + (\omega-1)^2}$.

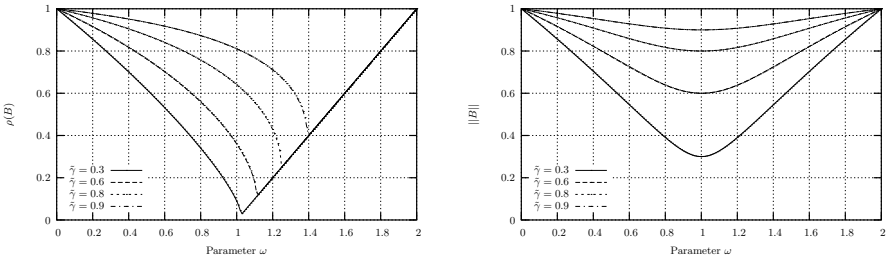
An abstract analysis of the spectral properties of the iteration operator B leads to the following result:

[3] An extension to a method using several patches has been analysed in [6].

Proposition 3.

1. *If* $\omega \in (0;2)$*, then Algorithm 2 converges, i.e.* $\lim_{n\to\infty} ||u^n - u_{Hh}|| = 0$.
2. *The spectral norm of* B *induced by the scalar product* $a(\cdot,\cdot)$ *is given by* $\|B\| = N(\tilde{\gamma},\omega) < 1$*, when* $\omega \in (0;2)$.
3. *The spectral radius of* B *is given by* $\rho(B) = \rho(\tilde{\gamma},\omega) < 1$*, when* $\omega \in (0;2)$.

Thus we have the convergence of Algorithm 2 when $\omega \in (0;2)$, the convergence speed given by $\rho(B)$, and the factor of the reduction of the error in the norm $a(\cdot,\cdot)^{1/2}$ bounded by $||B||$. Both functions are plotted on the graphs of Fig 2. In the case $V_0 = \{0\}$, $\tilde{\gamma}$ corresponds to the constant of the strengthened Cauchy-Buniakowski-Schwarz inequality.



(a) $\rho(\omega)$ for different $\tilde{\gamma}$. (b) $||B||$ for different $\tilde{\gamma}$.

Fig. 2. Spectral radius and norm of B as a function of ω for different γ.

Remark that in [3], Bramble et al. present an abstract analysis of product iterative methods and conclude with an upper bound for the norm of B. Even an optimization of the constants appearing in this bound (see [6]) shows that the estimate is always not optimal. We also point out that the minimization of this known result with respect to ω does not lead to a significative value for the relaxation parameter. We show that the best convergence speed, i.e. a minimal spectral radius (5), is obtained for $\omega = \omega_0(\tilde{\gamma})$ given by (6).

Let us briefly consider a case where $\overline{\Lambda} \subset K$, for $K \in \mathcal{T}_H$ and $r = 1$. Let the scalar product be given by

$$a(\psi,\varphi) = \sum_{i,j=1}^{d} \int_{\Omega} a_{ij} \frac{\partial \psi}{\partial x_j} \frac{\partial \varphi}{\partial x_i} \, \mathrm{d}\mathbf{x}, \quad \forall \psi, \varphi \in H_0^1(\Omega), \tag{7}$$

where $a_{ij} \in L^\infty(\Omega)$, $a_{ij}(x) = a_{ji}(x)$, $1 \le i,j \le d$, and $\sum_{i,j=1}^{d} a_{ij}(x)\xi_i\xi_j \ge \alpha|\xi|^2, \forall \xi \in \mathbb{R}^d, \forall x \in \Omega$. Set $\beta = \left[\sum_{j=1}^{d} \left(\sum_{i=1}^{d} ||\partial a_{ij}/\partial x_i||_{L^\infty(\Lambda)}\right)^2\right]^{\frac{1}{2}}$, and

$\delta = \sqrt{1/\tilde{\lambda}}$, $\tilde{\lambda}$ being the Poincaré constant. In this case we have $\tilde{\gamma} \leq \frac{\beta\delta}{\alpha}$, i.e. an upper bound for the parameter $\tilde{\gamma}$. If furthermore the a_{ij}'s are constant over Λ, $1 \leq i, j \leq d$, this last result implies that the algorithm converges in only one iteration.

A crucial question for running the algorithm is to know how to choose the relaxation parameter ω. Since Prop. 3, if $\omega = 1$, we have $\rho(B) = \tilde{\gamma}^2$. Furthermore, we can prove that $\rho(B) = \lim_{n\to\infty} \sqrt[n]{||B^n u^0||}$. Hence, given an evaluation of $\tilde{\gamma}$, we obtain the optimal relaxation parameter $\omega^{\text{opt}} = \omega_0(\tilde{\gamma})$ given by the formula (6). The parameter is optimal in the sense that it gives the minimum value for $\rho(B)$ directly related to the speed of convergence.

In practice, we set $\omega = 1$ and $f \equiv 0$, and perform m steps of the algorithm to obtain some u^m. Following what precedes, we use the approximation $\rho = \sqrt[m]{||u^m||}$, and obtain with (6) and $\rho = \tilde{\gamma}^2$ that $\omega^{\text{opt}} = \frac{2-2\sqrt{1-\rho}}{\rho}$.

Finally, we consider Algorithm 2 with two relaxation parameters ω_h and ω_H such that $u^{n-\frac{1}{2}} = u^{n-1} + \omega_h w_h$ and $u^n = u^{n-\frac{1}{2}} + \omega_H w_H$. We can prove that the spectral radius of the corresponding iteration operator is minimum when $\omega_H = \omega_h = \omega_0(\tilde{\gamma})$.

4 Implementation issues

We discuss practical aspects to construct an efficient computer program for implementing Algorithm 2. Handling two domains with a priori non-conforming triangulations raises a couple of practical issues. At any stage the coarse and the fine parts of the solution u^n are stored separately, that is to say $u^{n-1} = u_H^{n-1} + u_h^{n-1}$ with $u_H^{n-1} \in V_H$, $u_h^{n-1} \in V_h$. We write the first step of the n-th iteration of the algorithm as follows:

Find $w_h \in V_h$ s.t. $a(w_h, \varphi) = \langle f|\varphi\rangle - a(u_H^{n-1}, \varphi) - a(u_h^{n-1}, \varphi), \forall \varphi \in V_h$.
Set $u_H^{n-\frac{1}{2}} = u_H^{n-1}$ and $u_h^{n-\frac{1}{2}} = u_h^{n-1} + \omega w_h$.

The same holds for the second step which writes out explicitly:

Find $w_H \in V_H$ s.t. $a(w_H, \varphi) = \langle f|\varphi\rangle - a(u_H^{n-\frac{1}{2}}, \varphi) - a(u_h^{n-\frac{1}{2}}, \varphi), \forall \varphi \in V_H$.
Set $u_H^n = u_H^{n-\frac{1}{2}} + \omega w_H$ and $u_h^n = u_h^{n-\frac{1}{2}}$.

We conclude that $u_h^n = u_h^{n-1} + \omega w_h$ and $u_H^n = u_H^{n-1} + \omega w_H$.

At this point we need to discuss the numerical integration and restrict ourselves to linear finite elements ($r = s = 1$).

Two difficulties are to be taken into account whether sharp data, i.e. data needing fine integration, of the problem comes from the right-hand side f or originates from the form a. In the first case the evaluation of $\langle f|\varphi\rangle$ needs particular attention. In the second case scalar products evaluated on the coarse

grid must be considered with care. Another issue is the treatment of mixed term scalar products wherein both coarse and fine functions appear.

In the sequel, we consider these problems and illustrate our proposals with the scalar product given by (7). The evaluation of the different terms appearing in the algorithm is conforming to the following guidelines:

- If the coefficients a_{ij} defining the scalar product a are smooth in Λ, the homogeneous terms $a(\varphi_H, \psi_H)$ with $\varphi_H, \psi_H \in V_H$, and $a(\varphi_h, \psi_h)$ with $\varphi_h, \psi_h \in V_h$, of support in Ω resp. Λ are integrated using the grid $\mathcal{T}_H$ on Ω resp. $\mathcal{T}_h$ in Λ. Numerical integration in 2D is done with the standard three-point formula (in 3D we use a four-point formula). In the case of (7) this writes out, $\forall \varphi_H, \psi_H \in V_H$,

$$a(\varphi_H, \psi_H) \approx \sum_{K \in \mathcal{T}_H} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} \sum_{i,j=1}^{d} a_{ij}(\mathbf{x}_K^\alpha) \left. \frac{\partial \varphi_H}{\partial x_j} \right|_K \left. \frac{\partial \psi_H}{\partial x_i} \right|_K , \qquad (8)$$

  where $|K|$ denotes the area or volume, and $\mathbf{x}_K^\alpha$, $\alpha = 1, \ldots, d+1$, the vertices of the element K. We use the same formula for $a(\varphi_h, \psi_h)$ where $\varphi_h, \psi_h \in V_h$ with $K \in \mathcal{T}_h$ in (8).
  The mixed term $a(\varphi_h, \psi_H)$, $\varphi_h \in V_h$,$\psi_H \in V_H$, of support in Λ, is approximated by $a(\varphi_h, r_h \psi_H)$, i.e.

$$a(\varphi_h, \psi_H) \approx \sum_{K \in \mathcal{T}_h} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} \sum_{i,j=1}^{d} a_{ij}(\mathbf{x}_K^\alpha) \left. \frac{\partial \varphi_h}{\partial x_j} \right|_K \left. \frac{\partial (r_h \psi_H)}{\partial x_i} \right|_K , \qquad (9)$$

  where r_h is the standard interpolant to the space V_h. When implementing, we need to introduce a transmission grid, i.e. a fine structured grid considered over the patch Λ. This enables handling of the grids and associating fine and coarse triangles and vertices.
- If the coefficients a_{ij} are sharp in Λ, the presented approximation illustrated by (8) for the term $a(u_H^{n-\frac{1}{2}}, \varphi)$, $\varphi \in V_H$, appearing in the right-hand side of the coarse correction step needs to be rewritten in order to use a fine integration in the domain Λ. Set a_{ij}^1 and a_{ij}^2 such that $a_{ij} = a_{ij}^1 + a_{ij}^2$ and

$$a_{ij}^1 = \begin{cases} a_{ij} & \text{in } \Omega \setminus \Lambda \\ 0 & \text{in } \Lambda \end{cases} , \qquad a_{ij}^2 = \begin{cases} 0 & \text{in } \Omega \setminus \Lambda \\ a_{ij} & \text{in } \Lambda \end{cases} .$$

  The right-hand side of relation (8) rewrites, $\forall \varphi_H, \psi_H \in V_H$,

$$\begin{aligned} &\sum_{K \in \mathcal{T}_H} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} \sum_{i,j=1}^{d} a_{ij}^1(\mathbf{x}_K^\alpha) \left. \frac{\partial \varphi_H}{\partial x_j} \right|_K \left. \frac{\partial \psi_H}{\partial x_i} \right|_K \\ &+ \sum_{K \in \mathcal{T}_h} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} \sum_{i,j=1}^{d} a_{ij}^2(\mathbf{x}_K^\alpha) \left. \frac{\partial (r_h \varphi_H)}{\partial x_j} \right|_K \left. \frac{\partial (r_h \psi_H)}{\partial x_i} \right|_K . \qquad (10) \end{aligned}$$

As our algorithm is a correction algorithm with corrections tending to zero, the left-hand side $a(w_H, \varphi)$, $\varphi \in V_H$, is not to be rewritten. All other terms already based on $\mathcal{T}_h$ for integration do not need to be revised.

- The term $\langle f|\varphi\rangle$, $\varphi \in V_h$ or V_H, is approximated with

$$\langle f|\varphi_H\rangle \approx \sum_{K\in\mathcal{T}_H} \frac{|K|}{d+1}\sum_{\alpha=1}^{d+1} f^1(\mathbf{x}_K^\alpha)\varphi_H(\mathbf{x}_K^\alpha) + \sum_{K\in\mathcal{T}_h} \frac{|K|}{d+1}\sum_{\alpha=1}^{d+1} f^2(\mathbf{x}_K^\alpha)(r_h\varphi_H)(\mathbf{x}_K^\alpha), \quad \forall \varphi_H \in V_H, \quad (11)$$

and

$$\langle f|\varphi_h\rangle \approx \sum_{K\in\mathcal{T}_h} \frac{|K|}{d+1}\sum_{\alpha=1}^{d+1} f^2(\mathbf{x}_K^\alpha)\varphi_h(\mathbf{x}_K^\alpha), \quad \forall \varphi_h \in V_h, \quad (12)$$

where $f = f^1 + f^2$ with $f^1 = \begin{cases} f & \text{in } \Omega\setminus\Lambda \\ 0 & \text{in } \Lambda \end{cases}$, and $f^2 = \begin{cases} 0 & \text{in } \Omega\setminus\Lambda \\ f & \text{in } \Lambda \end{cases}$.

5 Applications in 2D and 3D

We consider the Poisson-Dirichlet problem

$$\begin{cases} -\Delta u = f & \text{in } \Omega = (-1;1)^d,\ d = 2,3, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \quad (13)$$

First, we implement the problem (13) in 2D ($d = 2$) to assess the convergence of Algorithm 2 with regard to the influence of the grids used. We take f such that the exact solution to the problem is given by $u = u_0 + \sum_{i=1}^4 u_i$, $u_0(x,y) = \cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y)$ and $u_i(x,y) = \eta\chi(R_i)\exp\epsilon_f^{-2}\exp(-1/|\epsilon_f^2 - R_i^2|)$, where $R_i(x,y) = \sqrt{(x-x_i)^2+(y-y_i)^2}$ and $\chi(R_i) = 1$ if $R_i \le \epsilon_f$, $\chi(R_i) = 0$ if $R_i > \epsilon_f$; η, ϵ_f and (x_i, y_i), $i = 1,2,3,4$ are parameters. Hence the right-hand side of (13) is given by $f = f_0 + \sum_{i=1}^4 f_i$, where $f_0 = -\Delta u_0$ and $f_i = -\Delta u_i$, $i = 1,2,3,4$. We choose $\eta = 10$, $\epsilon_f = 0.3$ and $(x_1, y_1) = (0.3, 0.3)$, $(x_2, y_2) = (0.7, 0.3)$, $(x_3, y_3) = (0.3, 0.7)$, $(x_4, y_4) = (0.7, 0.7)$.

For the triangulation of $\overline{\Omega}$, we use a coarse uniform grid with mesh size H and $r = 1$. We consider the patches Λ_i, $i = 1,2,3,4$, with a fine uniform triangulation of size h and $s = 1$. Choose $\Lambda_i = (x_i - \epsilon; x_i + \epsilon) \times (y_i - \epsilon; y_i + \epsilon)$, with $\epsilon = 0.1$. We set $H = 2/N$ and $h = 2\epsilon/M$, N, M being the number of discretization points on one side of the squares Ω and Λ_i respectively.

In the following we consider different situations including structured nested and non-nested as well as unstructured grids on the domain Ω. We always use the same structured grids for the patches. Our goal is to show that the algorithm performs well when $h \to 0$ for fixed H, and when each patch covers only a small number of coarse elements. It is particularly competitive when used

with the optimal relaxation parameter in initially ill-conditioned situations (see Table 1(c), with small displacement of the nodes of the nested grid).

We introduce a stopping criterium for the algorithm, which controls the relative discrepancy $||u^n - u^{n-1}||/||u^n||$ between two iterations $n-1$ and n, $n = 1, 2, \ldots$, and measures the stagnation of the algorithm. We call n_{cvg} the number of iterations required for convergence. Conforming to our problem (13), $||\cdot||$ denotes here the H^1-seminorm.

All results are illustrated in the following table. In each part we depict the considered situation by small graphics showing first the whole triangulation $\mathcal{T}_H$ with the patches, then a zoom to emphasize the region around one corner of a patch to show how $\mathcal{T}_h$ and $\mathcal{T}_H$ are related. First we set $\omega = 1$ and run our method to obtain an estimate of $\tilde{\gamma}$ and hence of the spectral radius of the iteration operator, as discussed at the end of Sect. 3. Then we run the algorithm on problem (13) till convergence and report the number of iterations n_{cvg}. These values are respectively reported in the first rows of Tables 1(a)–1(c). Given the approximation for $\tilde{\gamma}$ we determine the optimal relaxation parameter with (6) and give the spectral radius. The last line in the tables reports the required iterations needed by the method to converge under optimal relaxation.

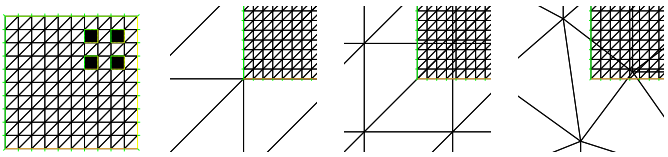
In a first test, we choose N and M such that the ratio H/h is of magnitude 10. In these first cases, the patches cover a small number of triangles of $\mathcal{T}_H$ leading to small coefficients $\tilde{\gamma}$ and ρ. Hence convergence is reached after a small number of iterations.

When doubling the number of fine triangles, see Table 1(b), the situation remains similar. A slight over-relaxation realises a gain of a couple of iterations. This suggests that the method is efficient in multi-scale situations, i.e. in problems with fixed H and $h \to 0$.

In the examples of Table 1(c) we increase the precision of the coarse triangulation. These cases show that the algorithm is best-suited to situations with patches covering a small number of coarse triangles. In fact, increasing the number of coarse triangles covered by the patches leads to bad condition numbers (ρ close to 1). Nevertheless optimal relaxation allows to divide by a factor 2 the number of iterations necessary to obtain convergence. This shows that optimal relaxation is a key ingredient in our method.

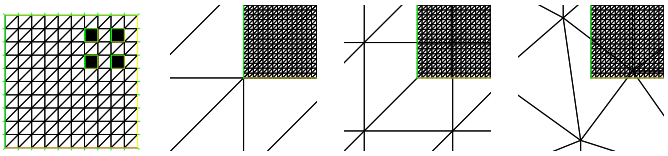
These basic results show that the method is very well adapted for multi-scale situations when applying small patches in the regions with sharp data.

Let us now turn to the 3D case ($d = 3$) of problem (13). We take f such that the exact solution to the problem is given by $u = u_0 + u_1$, $u_0(x, y, z) = \cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y)\cos(\frac{\pi}{2}z)$ and $u_1(x, y, z) = \eta\chi(R)\exp\epsilon_f^{-2}\exp(-1/|\epsilon_f^2 - R^2|)$. where $R(x, y, z) = \sqrt{x^2 + y^2 + z^2}$. We choose $\eta = 10$, $\epsilon_f = 0.3$ and take $\Lambda = (-0.25, 0.25)^3$. We set $\omega = 1$. For the triangulation of $\overline{\Omega}$ resp. $\overline{\Lambda}$, we use a uniform structured grid with mesh size H resp. h. We set $H = 2/N$ and $h = 0.5/M$, N, M being the number of points per side of the cubes Ω and



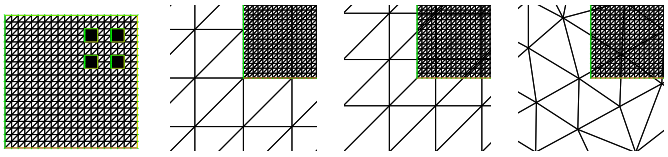
$H/h = 10$	nested $N = M = 10$	non-nested $N = 11,\ M = 10$	unstructured $N = M = 10$
$\rho(\tilde{\gamma}, 1) = \tilde{\gamma}^2$	0.28	0.30	0.34
n_{cvg}	6	8	8
$\rho(\tilde{\gamma}, \omega^{\text{opt}}) = \omega^{\text{opt}} - 1$	0.08	0.09	0.10
n_{cvg}	5	6	8

(a) $H/h = 10$ and $N = 10$.



$H/h = 20$	nested $N = 10,\ M = 20$	non-nested $N = 11,\ M = 20$	unstructured $N = 10,\ M = 20$
$\rho(\tilde{\gamma}, 1) = \tilde{\gamma}^2$	0.28	0.31	0.38
n_{cvg}	6	8	9
$\rho(\tilde{\gamma}, \omega^{\text{opt}}) = \omega^{\text{opt}} - 1$	0.08	0.09	0.12
n_{cvg}	5	6	6

(b) $H/h = 20$ and $N = 10$.



$H/h = 10$	nested $N = M = 20$	non-nested $N = 21,\ M = 20$	unstructured $N = M = 20$
$\rho(\tilde{\gamma}, 1) = \tilde{\gamma}^2$	0.24	0.89	0.91
n_{cvg}	6	24	27
$\rho(\tilde{\gamma}, \omega^{\text{opt}}) = \omega^{\text{opt}} - 1$	0.07	0.50	0.54
n_{cvg}	5	13	15

(c) $H/h = 20$ and $N = 20$.

Table 1. Comparison of the algorithm properties in 2D.

Λ. We use linear finite elements ($r = s = 1$). To assess the convergence of $u_{Hh} = u^{n_{\text{cvg}}}$ in H and h to the exact solution u,[4] we introduce the standard relative errors $e^n = ||u - u^n||/||u||$ and $e_{Hh} = e^{n_{\text{cvg}}} = ||u - u_{Hh}||/||u||$.

Consider the coarse triangulation ($N = 16, 32, 64$) with a patch $M = 8, 16, 32, 64$. We assess the quality of the estimate u^n at the iteration n of the algorithm by comparing it to the exact solution u. The results of e^n through n are depicted on Fig 3(a). Note that it is useful to run the algorithm through more than one sole iteration. Nevertheless only a couple of iterations are sufficient to obtain good results. As mentioned above, in the present cases the speed of convergence remains constant with respect to the refinement of the patch. When the error in $\Omega \setminus \Lambda$ dominates (case $N = 16$, $M = 32, 64$) a refinement of $\mathcal{T}_h$ does not improve the precision. The reduction of the error, in comparison with the sequence $M = 8$ to $M = 16$, stagnates.

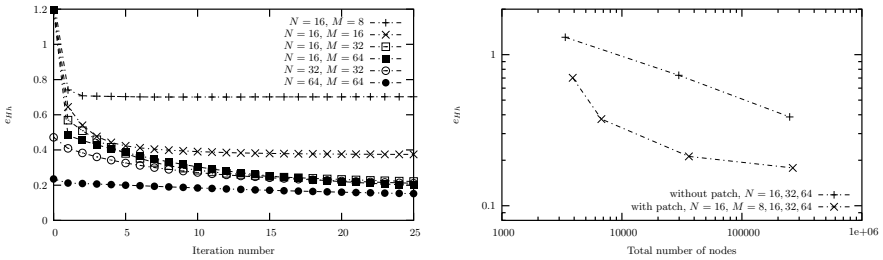
Let us illustrate the efficiency of the method with respect to the memory usage. On one hand we consider the computation of u_H on one grid with $N = 16, 32, 64$. On the other hand we take a coarse grid ($N = 16$) with a fine grid in the patch $M = 8, 16, 32, 64$. In Fig. 3(b) we plot the error e_{Hh} with regard to the number of nodes used. Comparison of both curves leads to conclude that the method is efficient in terms of memory usage. As above, the stagnation in the reduction of e_{Hh} stems from the error on the coarse grid becoming dominant. Similar results to those of memory usage can obtained for the CPU-time.

In Fig. 3(c) we illustrate the solution obtained after 5 iterations for the test case $N = 16$, $M = 32$.

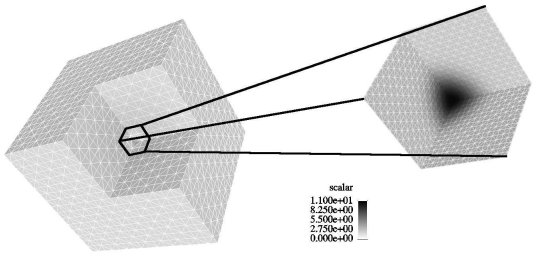
References

1. Y. Achdou and Y. Maday. The mortar element method with overlapping subdomains. *SIAM J. Numer. Anal.*, 40(2):601–628, 2002.
2. R.E. Bank, T.F. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numer. Math.*, 52:427–458, 1988.
3. J.H. Bramble, J.E. Pasciak, J. Wang, and J. Xu. Convergence estimates for product iterative methods with applications to domain decomposition. *Math. Comp.*, 57(195):1–21, 1991.
4. F. Brezzi, J.-L. Lions, and O. Pironneau. Analysis of a chimera method. *C. R. Acad. Sci. Paris, Ser. I*, 332:655–660, 2003.
5. T.F. Chan, B.F. Smith, and J. Zou. Overlapping Schwarz methods on unstructured meshes using non-matching coarse grids. *Numer. Math.*, 73:149–167, 1996.
6. R. Glowinski, J. He, A. Lozinski, J. Rappaz, and J. Wagner. Finite element approximation of multi-scale elliptic problems using patches of elements. *Submitted to Numer. Math.*, 2004.

[4] An assessment of the convergence in H and h illustrating the *a priori* estimate (3) is given in [6], Fig. 6.



(a) e^n versus iteration number. (b) e_{Hh} versus number of nodes.



(c) Estimate u^5 for $N = 16$, $M = 32$.

Fig. 3. Results in 3D and illustrations.

7. R. Glowinski, J. He, J. Rappaz, and J. Wagner. Approximation of multi-scale elliptic problems using patches of finite elements. *C. R. Acad. Sci. Paris, Ser. I*, 337:679–684, 2003.
8. S.F. McCormick and J.W. Ruge. Unigrid for multigrid simulation. *Math. Comp.*, 41(163):43–62, 1983.
9. S.F. McCormick and J. Thomas. The Fast Adaptive Composite grid (FAC) method for elliptic equations. *Math. Comp.*, 46(174):439–456, 1986.
10. H.A. Schwarz. *Ueber einen Grenzübergang durch alternirendes Verfahren*, volume 2 of *Gesammelte Mathematische Abhandlungen*, pages 133–143. Chelsea Publishing Company, Bronx, New York, 1972.
11. J.L. Steger and J.A. Benek. On the use of composite grid schemes in computational aerodynamics. *Comput. Methods Appl. Mech. Engrg.*, 64:301–320, 1987.
12. J. Xu and L. Zikatanov. The method of alternating projections and the method of subspace corrections in Hilbert space. *PennState, Department of Mathematics*, AM223, 2000.
13. H. Yserentant. On the multi-level splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.