How to Use the Smith Factorization for Domain Decomposition Methods Applied to the Stokes Equations

Victorita Dolean[1], Frédéric Nataf[2], and Gerd Rapin[3]

[1] Univ. de Nice Sophia-Antipolis, Laboratoire J.-A. Dieudonné, Nice, France. dolean@math.unice.fr
[2] Laboratoire J. L. Lions, Université Pierre et Marie Curie, 75252 Paris Cedex 05, France. nataf@ann.jussieu.fr
[3] Math. Dep., NAM, University of Göttingen, D-37083, Germany. grapin@math.uni-goettingen.de

Summary. In this paper we demonstrate that the Smith factorization is a powerful tool to derive new domain decomposition methods for vector valued problems. Here, the factorization is applied to the two-dimensional Stokes system. The key idea is the transformation of the Stokes problem into a scalar bi-harmonic problem. We show how a proposed domain decomposition method for the bi-harmonic problem leads to an algorithm for the Stokes equations which inherits the convergence behavior of the scalar problem.

1 Introduction

The last decade has shown that Neumann-Neumann type algorithms, FETI, and BDDC methods are very efficient domain decomposition methods for scalar symmetric positive definite second order problems. Then, these methods have been extended to other problems, like advection-diffusion equations, plate or shell problems. Also for the Stokes equations several iterative substructuring methods have been discussed in the literature, like Neumann-Neumann precondtioners (cf. [6, 2]), FETI (cf. [3]) or BDDC methods (cf. [4]).

Our work is motivated by the fact that many domain decomposition methods for vector valued problems are less optimal than domain decomposition methods for scalar problems. Indeed, in the case of two subdomains consisting of the two half planes it is well known that Neumann-Neumann preconditioners are exact (the preconditioned operator simplifies to the identity) preconditioners for the Schur complement equation for scalar equations like the Laplace problem. Unfortunately, this is not valid for the Stokes problem as we have shown in [5] for standard Neumann-Neumann preconditioners. The goal of this paper is the derivation of an algorithm which preserves this property, cf. [1] for detailed proofs.

Using the Smith factorization we show the equivalence between the Stokes equations and a bi-harmonic problem in Section 2. The Smith factorization is a classical algebraic tool for matrices with polynomial entries. Then, in Section 3 we introduce an exact domain decomposition method for the bi-harmonic equation and transform it to the Stokes equations. Section 4 is dedicated to numerical results. Finally, we give some concluding remarks.

2 Equivalence Between the Stokes Equations and Bi-harmonic Problems

We will show the equivalence between the two-dimensional Stokes system

$$-\nu \varDelta \boldsymbol{u} + \nabla p + c\boldsymbol{u} = \boldsymbol{f}, \quad \nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \varOmega$$

and a fourth order scalar problem (the bi-harmonic problem) by means of the Smith factorization. This is motivated by the fact that scalar problems are easier to manipulate and the construction of new algorithms is more intuitive. The approach is not limited to the two-dimensional case. The three-dimensional case is discussed in [1].

The data is given by $\boldsymbol{f} = (f_1, f_2)^T \in [L^2(\varOmega)]^2$, $\nu > 0$, and $c \geq 0$. Very often c stems from an implicit time discretization and then c is given by the inverse of the time step size. We denote the two-dimensional Stokes operator by $\mathcal{S}_2(\boldsymbol{v}, q) := -\nu\varDelta\boldsymbol{v} + c\boldsymbol{v} + \nabla q$. We recall the Smith factorization of a matrix with polynomial entries ([7], Theorem 1.4):

Theorem 1. *Let A be a $n \times n$ matrix with polynomial entries with respect to the variable λ: $A = (a_{ij}(\lambda))_{1\leq i,j \leq n}$. Then, there exist matrices E, D and F with polynomial entries satisfying the following properties:*

- *$det(E)$ and $det(F)$ are constants,*
- *D is a diagonal matrix uniquely determined up to a multiplicative constant,*
- *$A = EDF$.*

The Smith factorization is applied to the two-dimensional model problem $\mathcal{S}_2(\boldsymbol{u}, p) = \boldsymbol{g}$ in $\mathbb{R}^2$ with right hand side $\boldsymbol{g} = (f_1, f_2, 0)^T$ where we suppose that all variables vanish at infinity. Moreover, it is assumed that the coefficients c, ν are constants. The spatial coordinates are denoted by x and y. In order to apply the factorization to the Stokes system we first take formally the Fourier transform of $\mathcal{S}_2(\boldsymbol{u}, p) = \boldsymbol{g}$ with respect to y. The dual variable is denoted by k. The Fourier transform of a function f is written as $\hat{f}$ or $\mathcal{F}_y f$. Thus, we get

$$\hat{\mathcal{S}}_2(\hat{\boldsymbol{u}}, \hat{p}) = \begin{pmatrix} -\nu(\partial_{xx} - k^2) + c & 0 & \partial_x \\ 0 & -\nu(\partial_{xx} - k^2) + c & ik \\ \partial_x & ik & 0 \end{pmatrix} \begin{pmatrix} \hat{u} \\ \hat{v} \\ \hat{p} \end{pmatrix}. \tag{1}$$

Considering $\hat{\mathcal{S}}_2(\hat{\boldsymbol{u}}, \hat{p})$ as a matrix with polynomial entries with respect to ∂_x we perform for $k \neq 0$ the Smith factorization. We obtain

$$\hat{\mathcal{S}}_2 = \hat{E}_2 \hat{D}_2 \hat{F}_2 \tag{2}$$

with a diagonal matrix $\hat{D}_2 = \text{diag}(1, 1, (\partial_{xx} - k^2)\hat{\mathcal{L}}_2)$ and

$$\hat{F}_2 = \begin{pmatrix} \nu k^2 + c & \nu i k \partial_x & \partial_x \\ 0 & \hat{\mathcal{L}}_2 & ik \\ 0 & 1 & 0 \end{pmatrix}, \quad \hat{E}_2 = \hat{T}_2^{-1} \begin{pmatrix} ik\hat{\mathcal{L}}_2 & \nu\partial_{xxx} & -\nu\partial_x \\ 0 & \hat{T}_2 & 0 \\ ik\partial_x & -\partial_{xx} & 1 \end{pmatrix}$$

where T_2 is a differential operator in y-direction whose symbol is $ik(\nu k^2 + c)$. Moreover, $\hat{\mathcal{L}}_2 := \nu(-\partial_{xx} + k^2) + c$ is the Fourier transform of $\mathcal{L}_2 := -\nu\Delta + c$.

Remark 1. Thus, the Stokes problem $\mathcal{S}_2(\boldsymbol{u}, p) = \boldsymbol{g}$ in $\mathbb{R}^2$ can be written as

$$\hat{D}_2 \hat{\boldsymbol{w}} = \hat{E}_2^{-1} \hat{\boldsymbol{g}}, \qquad \hat{\boldsymbol{w}} := (\hat{w}_1, \hat{w}_2, \hat{w}_3)^T := \hat{F}_2(\hat{\boldsymbol{u}}, \hat{p})^T. \tag{3}$$

From (3) we get $\hat{w}_1 = (\hat{E}_2^{-1}\hat{\boldsymbol{g}})_1$ and $\hat{w}_2 = (\hat{E}_2^{-1}\hat{\boldsymbol{g}})_2$. Noticing that $\hat{w}_3 = \left(\hat{F}_2(\hat{\boldsymbol{u}}, \hat{p})^T\right)_3$ $= \hat{v}$ the previous equation yields after applying an inverse Fourier transform

$$\Delta(-\nu\Delta + c)v = \mathcal{F}_y^{-1}\left((\hat{E}_2^{-1}\hat{\boldsymbol{g}})_3\right). \tag{4}$$

Since the determinants of the matrices $\hat{E}_2$ and $\hat{F}_2$ are non-zero numbers (i.e. a polynomial of order zero) the entries of their inverses are still polynomial in ∂_x. Thus, applying $\hat{E}_2^{-1}$ to the right hand side $\hat{\boldsymbol{g}}$ amounts to taking derivatives of $\hat{\boldsymbol{g}}$ and making linear combinations of them. If the plane $\mathbb{R}^2$ is split into subdomains $\mathbb{R}^- \times \mathbb{R}$ and $\mathbb{R}^+ \times \mathbb{R}$ the application of $\hat{E}_2^{-1}$ and $\hat{F}_2^{-1}$ to a vector can be done for each subdomain independently. No communication between the subdomains is necessary. The local problems are only coupled by the biharmonic problem (4). Thus, we can obtain a domain decomposition method for the Stokes problem by defining a domain decomposition method for (4) and recasting it to the Stokes problem using the Smith factorization.

3 A New Algorithm for the Stokes Equations

We construct an algorithm for $\mathcal{B} := \Delta\mathcal{L}_2 = \Delta(-\nu\Delta + c)$ on the whole plane divided into two half-planes, which converges in two iterations. Then, via the Smith factorization, we recast it in a new algorithm for the Stokes system.

We consider the following problem: Find $\phi : \mathbb{R}^2 \to \mathbb{R}$ such that

$$\mathcal{B}(\phi) = g \text{ in } \mathbb{R}^2, \qquad |\phi(\boldsymbol{x})| \to 0 \text{ for } |\boldsymbol{x}| \to \infty \tag{5}$$

where g is a given right hand side. The domain $\Omega = \mathbb{R}^2$ is decomposed into two half planes $\Omega_1 = \mathbb{R}^- \times \mathbb{R}$ and $\Omega_2 = \mathbb{R}^+ \times \mathbb{R}$ with interface $\Gamma := \{0\} \times \mathbb{R}$. Let $(\boldsymbol{n}_i)_{i=1,2}$ be the outward normal of $(\Omega_i)_{i=1,2}$. In contrast to the overlapping additive Schwarz algorithm in [8] we propose an iterative-substructuring algorithm.

ALGORITHM 1 *For any initial values ϕ_1^0 and ϕ_2^0 with $\phi_1^0 = \phi_2^0$ and $\mathcal{L}_2\phi_1^0 = \mathcal{L}_2\phi_2^0$ on Γ we obtain $(\phi_i^{n+1})_{i=1,2}$ from $(\phi_i^n)_{i=1,2}$ by the following procedure:*

Correction step. *We compute the corrections $(\tilde{\phi}_i^{n+1})_{i=1,2}$:*

$$\begin{cases} \mathcal{B}\tilde{\phi}_1^{n+1} = 0 \text{ in } \Omega_1 \\ \lim\limits_{|\mathbf{x}|\to\infty} |\tilde{\phi}_1^{n+1}| = 0 \\ \dfrac{\partial\tilde{\phi}_1^{n+1}}{\partial\boldsymbol{n}_1} = \gamma_1^n \text{ on } \Gamma \\ \dfrac{\partial\mathcal{L}_2\tilde{\phi}_1^{n+1}}{\partial\boldsymbol{n}_1} = \gamma_2^n \text{ on } \Gamma \end{cases} \qquad \begin{cases} \mathcal{B}\tilde{\phi}_2^{n+1} = 0 \text{ in } \Omega_2 \\ \lim\limits_{|\mathbf{x}|\to\infty} |\tilde{\phi}_2^{n+1}| = 0 \\ \dfrac{\partial\tilde{\phi}_2^{n+1}}{\partial\boldsymbol{n}_2} = \gamma_1^n \text{ on } \Gamma \\ \dfrac{\partial\mathcal{L}_2\tilde{\phi}_2^{n+1}}{\partial\boldsymbol{n}_2} = \gamma_2^n \text{ on } \Gamma \end{cases} \tag{6}$$

where $\gamma_1^n = -\dfrac{1}{2}\left(\dfrac{\partial \phi_1^n}{\partial \boldsymbol{n}_1} + \dfrac{\partial \phi_2^n}{\partial \boldsymbol{n}_2}\right)$ *and* $\gamma_2^n = -\dfrac{1}{2}\left(\dfrac{\partial \mathcal{L}_2\phi_1^n}{\partial \boldsymbol{n}_1} + \dfrac{\partial \mathcal{L}_2\phi_2^n}{\partial \boldsymbol{n}_2}\right)$.

Updating step. *We update* $(\phi_i^{n+1})_{i=1,2}$ *by solving the local problems:*

$$\left\{\begin{array}{ll} \mathcal{B}\phi_1^{n+1} & = g \ \text{ in } \Omega_1 \\ \lim\limits_{|\mathbf{x}|\to\infty} |\phi_1^{n+1}| = 0 & \\ \phi_1^{n+1} & = \phi_1^n + \delta_1^{n+1} \ \text{ on } \Gamma \\ \mathcal{L}_2\phi_1^{n+1} & = \mathcal{L}_2\phi_1^n + \delta_2^{n+1} \ \text{ on } \Gamma \end{array}\right. \quad \left\{\begin{array}{ll} \mathcal{B}\phi_2^{n+1} & = g \ \text{ in } \Omega_2, \\ \lim\limits_{|\mathbf{x}|\to\infty} |\phi_2^{n+1}| = 0 & \\ \phi_2^{n+1} & = \phi_2^n + \delta_1^{n+1} \ \text{ on } \Gamma \\ \mathcal{L}_2\phi_2^{n+1} & = \mathcal{L}_2\phi_2^n + \delta_2^{n+1} \ \text{ on } \Gamma \end{array}\right. \tag{7}$$

where $\delta_1^{n+1} = \dfrac{1}{2}(\tilde{\phi}_1^{n+1} + \tilde{\phi}_2^{n+1})$ *and* $\delta_2^{n+1} = \dfrac{1}{2}(\mathcal{L}_2\tilde{\phi}_1^{n+1} + \mathcal{L}_2\tilde{\phi}_2^{n+1})$.

Using the Fourier transform we can prove the following result.

Proposition 1. *Algorithm 1 converges in two iterations.*

After having found an optimal algorithm which converges in two steps for the fourth order operator $\mathcal{B}$ problem we focus on the Stokes system. It suffices to replace the operator $\mathcal{B}$ by the Stokes system and ϕ by the last component $(F_2(\boldsymbol{u},p)^T)_3$ of the vector $F_2(\boldsymbol{u},p)^T$ in the boundary conditions.

ALGORITHM 2 *We choose* $(\boldsymbol{u}_1^0, p_1^0)$ *and* $(\boldsymbol{u}_2^0, p_2^0)$ *such that* $(F_2(\boldsymbol{u}_1^0, p_1^0)^T)_3 = (F_2(\boldsymbol{u}_2^0, p_2^0)^T)_3$ *and* $\mathcal{L}_2(F_2(\boldsymbol{u}_1^0, p_1^0)^T)_3 = \mathcal{L}_2(F_2(\boldsymbol{u}_2^0, p_2^0)^T)_3$ *on* Γ.
We compute $((\boldsymbol{u}_i^{n+1}, p_i^{n+1}))_{i=1,2}$ *from* $((\boldsymbol{u}_i^n, p_i^n))_{i=1,2}$ *by the following iterative procedure:*

Correction step. *We compute the corrections* $((\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}))_{i=1,2}$*:*

$$\left\{\begin{array}{ll} \mathcal{S}_2(\tilde{\boldsymbol{u}}_1^{n+1}, \tilde{p}_1^{n+1}) & = 0 \ \text{ in } \Omega_1 \\ \lim\limits_{|\boldsymbol{x}|\to\infty} |\tilde{\boldsymbol{u}}_1^{n+1}| & = 0 \\ \dfrac{\partial(F_2(\tilde{\boldsymbol{u}}_1^{n+1}, \tilde{p}_1^{n+1})^T)_3}{\partial \boldsymbol{n}_1} & = \gamma_1^n \ \text{ on } \Gamma \\ \dfrac{\partial\mathcal{L}_2(F_2(\tilde{\boldsymbol{u}}_1^{n+1}, \tilde{p}_1^{n+1})^T)_3}{\partial \boldsymbol{n}_1} & = \gamma_2^n \ \text{ on } \Gamma \end{array}\right. \quad \left\{\begin{array}{ll} \mathcal{S}_2(\tilde{\boldsymbol{u}}_2^{n+1}, \tilde{p}_2^{n+1}) & = 0 \ \text{ in } \Omega_2 \\ \lim\limits_{|\boldsymbol{x}|\to\infty} |\tilde{\boldsymbol{u}}_2^{n+1}| & = 0 \\ \dfrac{\partial(F_2(\tilde{\boldsymbol{u}}_2^{n+1}, \tilde{p}_2^{n+1})^T)_3}{\partial \boldsymbol{n}_2} & = \gamma_1^n \ \text{ on } \Gamma \\ \dfrac{\partial\mathcal{L}_2(F_2(\tilde{\boldsymbol{u}}_2^{n+1}, \tilde{p}_2^{n+1})^T)_3}{\partial \boldsymbol{n}_2} & = \gamma_2^n \ \text{ on } \Gamma \end{array}\right. \tag{8}$$

where

$$\begin{aligned} \gamma_1^n &= -\frac{1}{2}\left(\frac{\partial(F_2(\boldsymbol{u}_1^n, p_1^n)^T)_3}{\partial \boldsymbol{n}_1} + \frac{\partial(F_2(\boldsymbol{u}_2^n, p_2^n)^T)_3}{\partial \boldsymbol{n}_2}\right) \\ \gamma_2^n &= -\frac{1}{2}\left(\frac{\partial\mathcal{L}_2(F_2(\boldsymbol{u}_1^n, p_1^n)^T)_3}{\partial \boldsymbol{n}_1} + \frac{\partial\mathcal{L}_2(F_2(\boldsymbol{u}_2^n, p_2^n)^T)_3}{\partial \boldsymbol{n}_2}\right). \end{aligned}$$

Updating step. *We update* $((\boldsymbol{u}_i^{n+1}, p_i^{n+1}))_{i=1,2}$ *by solving the local problems:*

$$\left\{\begin{array}{ll} \mathcal{S}_2(\boldsymbol{u}_i^{n+1}, p_i^{n+1}) & = \boldsymbol{g} \ \text{ in } \Omega_i \\ \lim\limits_{|\boldsymbol{x}|\to\infty} |\boldsymbol{u}_i^{n+1}| & = 0 \\ (F_2(\boldsymbol{u}_i^{n+1}, p_i^{n+1})^T)_3 & = (F_2(\boldsymbol{u}_i^n, p_i^n)^T)_3 + \delta_1^{n+1} \ \text{ on } \Gamma \\ \mathcal{L}_2(F_2(\boldsymbol{u}_i^{n+1}, p_i^{n+1})^T)_3 & = \mathcal{L}_2(F_2(\boldsymbol{u}_i^n, p_i^n)^T)_3 + \delta_2^{n+1} \ \text{ on } \Gamma \end{array}\right. \tag{9}$$

where

$$\begin{aligned} \delta_1^{n+1} &= \frac{1}{2}[(F_2(\tilde{\boldsymbol{u}}_1^{n+1}, \tilde{p}_1^{n+1})^T)_3 + (F_2(\tilde{\boldsymbol{u}}_2^{n+1}, \tilde{p}_2^{n+1})^T)_3], \\ \delta_2^{n+1} &= \frac{1}{2}[\mathcal{L}_2(F_2(\tilde{\boldsymbol{u}}_1^{n+1}, \tilde{p}_1^{n+1})^T)_3 + \mathcal{L}_2(F_2(\tilde{\boldsymbol{u}}_2^{n+1}, \tilde{p}_2^{n+1})^T)_3]. \end{aligned}$$

This algorithm seems quite complex since it involves third order derivatives of the unknowns in the boundary conditions on $(F_2(\tilde{\boldsymbol{u}}_i, \tilde{p}_i)^T)_3$. Writing $\boldsymbol{u}_i = (u_i, v_i)$ and using $(F_2(\tilde{\boldsymbol{u}}_i, \tilde{p}_i)^T)_3 = \tilde{v}_i$ it is possible to simplify it. By using the Stokes equations in the subdomains we can lower the degree of the derivatives in the boundary conditions. We further introduce the stress

$$\boldsymbol{\sigma}^i(\boldsymbol{u}, p) := \nu \partial_{\boldsymbol{n}_i} \boldsymbol{u} - p \boldsymbol{n}_i$$

on the boundary $\partial \Omega_i$ for a velocity $\boldsymbol{u} = (u, v)$, a pressure p and the normal vector $\boldsymbol{n}_i$. For any vector $\boldsymbol{u}$ its normal (resp. tangential) component on the interface is $u_{\boldsymbol{n}_i} = \boldsymbol{u} \cdot \boldsymbol{n}_i$ (resp. $u_{\boldsymbol{\tau}_i} = (I - \boldsymbol{n}_i \otimes \boldsymbol{n}_i)\boldsymbol{u}$). We denote $\sigma^i_{\boldsymbol{n}_i} := \sigma^i_{\boldsymbol{n}_i}(\boldsymbol{u}_i, p_i) \cdot \boldsymbol{n}_i$ and $\sigma^i_{\boldsymbol{\tau}_i} := (I - \boldsymbol{n}_i \otimes \boldsymbol{n}_i)\sigma^i$ as the normal and tangential parts of $\boldsymbol{\sigma}^i$, respectively. We can thus write the new algorithm for the Stokes equations for general decomposition into non overlapping subdomains: $\bar{\Omega} = \cup_{i=1}^N \bar{\Omega}_i$ and denote by Γ_{ij} the interface between subdomains Ω_i and Ω_j, $i \neq j$. The new algorithm for the Stokes system reads:

ALGORITHM 3 *Starting with an initial guess* $((\boldsymbol{u}_i^0, p_i^0))_{i=0}^N$ *satisfying* $\boldsymbol{u}^0_{i,\boldsymbol{\tau}_i} = \boldsymbol{u}^0_{j,\boldsymbol{\tau}_j}$ *and* $\sigma^i_{\boldsymbol{n}_i}(\boldsymbol{u}_i^0, p_i^0) = \sigma^j_{\boldsymbol{n}_j}(\boldsymbol{u}_j^0, p_j^0)$ *on* Γ_{ij}, $\forall i, j$, $i \neq j$, *the* **correction step** *is expressed as follows for* $1 \leq i \leq N$*:*

$$\begin{cases} \mathcal{S}_2(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) & = 0 \quad \text{in } \Omega_i \\ \tilde{u}^{n+1}_{i,\boldsymbol{n}_i} & = -\dfrac{1}{2}(u^n_{i,\boldsymbol{n}_i} + u^n_{j,\boldsymbol{n}_j}) \quad \text{on } \Gamma_{ij} \\ \boldsymbol{\sigma}^i_{\boldsymbol{\tau}_i}(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) & = -\dfrac{1}{2}(\boldsymbol{\sigma}^i_{\boldsymbol{\tau}_i}(\boldsymbol{u}_i^n, \tilde{p}_i^n) + \boldsymbol{\sigma}^j_{\boldsymbol{\tau}_j}(\boldsymbol{u}_j^n, \tilde{p}_j^n)) \quad \text{on } \Gamma_{ij} \end{cases} \tag{10}$$

followed by an **updating step** *for* $1 \leq i \leq N$*:*

$$\begin{cases} \mathcal{S}_2(\boldsymbol{u}_i^{n+1}, p_i^{n+1}) & = \boldsymbol{g} \quad \text{in } \Omega_i \\ \boldsymbol{u}^{n+1}_{i,\boldsymbol{\tau}_i} & = \boldsymbol{u}^n_{i,\boldsymbol{\tau}_i} + \dfrac{1}{2}(\tilde{\boldsymbol{u}}^{n+1}_{i,\boldsymbol{\tau}_i} + \tilde{\boldsymbol{u}}^{n+1}_{j,\boldsymbol{\tau}_j}) \quad \text{on } \Gamma_{ij} \\ \sigma^i_{\boldsymbol{n}_i}(\boldsymbol{u}_i^{n+1}, p_i^{n+1}) & = \sigma^i_{\boldsymbol{n}_i}(\boldsymbol{u}_i^n, p_i^n) \\ & \quad + \dfrac{1}{2}(\sigma^i_{\boldsymbol{n}_i}(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) + \sigma^j_{\boldsymbol{n}_j}(\tilde{\boldsymbol{u}}_j^{n+1}, \tilde{p}_j^{n+1})) \text{ on } \Gamma_{ij}. \end{cases} \tag{11}$$

Since Algorithm 3 is only a reformulation of Algorithm 1 we obtain:

Proposition 2. *For a domain* $\Omega = \mathbb{R}^2$ *divided into two non overlapping half planes, Algorithms 2 and 3 are equivalent and converge in two iterations.*

In each iteration step of Algorithm 3 two local boundary value problems have to be solved in each subdomain. Therefore the cost of an iteration step is the same as for the NN algorithm.

4 Numerical Results

For the discretization of the two-dimensional case we choose a second order centered Finite Volume approach with a staggered grid. We consider two different types of domain decomposition methods: the discrete version of Algorithm 3 and an accelerated version using the GMRES method.

In the sequel we compare the performance of the new algorithm with the standard Schur complement approach using a Neumann-Neumann preconditioner (without coarse space), cf. [2]. We consider the domain $\Omega = [0.2, 1.2] \times [0.1, 1.1]$. We choose

$\nu = 1$ and the right hand side $\boldsymbol{f}$ such that the exact solution $\boldsymbol{u} = (u, v)$ is given by $u(x, y) = \sin(\pi x)^3 \sin(\pi y)^2 \cos(\pi y)$, $v(x, y) = -\sin(\pi x)^2 \sin(\pi y)^3 \cos(\pi x)$ and $p(x, y) = x^2 + y^2$.

First, the interface system is solved by a purely iterative method (denoted respectively by it_{new} and it_{NN} for the new algorithm and the Neumann-Neumann preconditioner) and then accelerated by GMRES (denoted respectively by ac_{New} and ac_{NN}). In all tables we count the smallest number of iterations, which is needed to reduce the euclidian norm of the residual by $TOL = 10^{-8}$. In brackets the number of steps is printed, which is needed to achieve an error with respect to one-domain solution which is less than 10^{-6}. The case that the method is not converged within 100 steps is denoted by $-$.

We first consider a decomposition into two subdomains of same width and study the influence of the reaction parameter and of the mesh size on the convergence. We can see in Table 1 (left) that the convergence of the new algorithm is optimal. For the iterative version convergence is reached in two iterations. Since in this case the preconditioned operator for the corresponding Krylov method reduces in theory to the identity, the Krylov method converges in one step. This is also valid numerically. Moreover, both algorithm are completely insensitive with respect to the reaction parameter. The advantage in comparison to the Neumann-Neumann algorithm is obvious.

In Table 1 (right) we fix the reaction parameter $c = 10^{-5}$ and vary the mesh size: Both algorithms converge independently of the mesh size and, again, we observe a clearly better convergence behavior of the new algorithm. The same kind of results are valid for different values of c (not presented here).

Table 1. Influence of the reaction parameter on the convergence ($h = \frac{1}{96}$) (left), influence of the mesh size for $c = 10^{-5}$ (right).

c	it_{New}	it_{NN}	ac_{New}	ac_{NN}
10^2	2 (2)	16 (15)	1 (1)	6 (6)
10^0	2 (2)	17 (15)	1 (1)	6 (6)
10^{-3}	2 (2)	17 (15)	1 (1)	6 (6)
10^{-5}	2 (2)	17 (15)	1 (1)	6 (6)

h	it_{New}	it_{NN}	ac_{New}	ac_{NN}
1/24	2 (2)	16 (14)	1 (1)	6 (6)
1/48	2 (2)	17 (15)	1 (1)	6 (6)
1/96	2 (2)	17 (15)	1 (1)	6 (6)

Now, the case of a strip-wise decomposition into more than two subdomains is considered. The mesh size is fixed ($h = 1/96$) and for different values of c we vary the number of subdomains. In the case of a strip-wise decomposition into N subdomains, the iteration number is increasing very quickly for very small c and in Table 2 (left) we can see only a small advantage of the new algorithm over the more classical approach. For larger c (Table 2 (right)) the behavior of the two domain case is conserved. The number of iteration steps is almost reduced by a factor of two. Moreover, for all cases the convergence is still independent of the mesh size.

The final test cases treat general decompositions into $N \times N$ subdomains. Two different values for the reaction coefficient c are analyzed. The iterative variants do not converge in the multi-domain case with cross points within 100 steps (except one case), cf. Table 3. Applying the accelerated variants we observe in the case 2×2

Table 2. Influence of the number of subdomains ($h = \frac{1}{96}$): $c = 10^{-5}$ (left), $c = 10^2$ (right).

N	it_{New}	it_{NN}	ac_{New}	ac_{NN}
2	2 (2)	17 (15)	1 (1)	6 (6)
4	- (-)	- (-)	6 (8)	7 (-)
6	- (-)	- (-)	10 (15)	13 (-)
8	- (-)	- (-)	13 (21)	19 (-)

N	it_{New}	it_{NN}	ac_{New}	ac_{NN}
2	2 (2)	16 (15)	1 (1)	6 (6)
4	45 (34)	- (-)	5 (5)	10 (9)
6	- (-)	- (-)	8 (7)	15 (15)
8	- (-)	- (-)	11 (10)	21 (21)

Table 3. Influence of the number of subdomains ($h = \frac{1}{96}$): $c = 1$ (left), $c = 10^2$ (right).

$N \times N$	it_{New}	it_{NN}	ac_{New}	ac_{NN}
2x2	- (-)	- (-)	9 (9)	13 (13)
3x3	- (-)	- (-)	27 (30)	26 (28)
4x4	- (-)	- (-)	35 (39)	36 (39)

$N \times N$	it_{New}	it_{NN}	ac_{New}	ac_{NN}
2x2	66 (61)	- (-)	8 (7)	11 (11)
3x3	- (-)	- (-)	21 (22)	21 (21)
4x4	- (-)	- (-)	25 (27)	27 (27)

a faster convergence of the new algorithm. For more subdomains both algorithms need almost the same number of iteration steps. This behavior can be explained by the presence of floating subdomains, which causes additional problems. Here, a suitable coarse space will decrease the number of needed iteration steps.

5 Conclusion

We have shown that the Smith factorization is a powerful tool in order to derive new domain decomposition methods for vector valued partial differential equations. The proposed algorithm for the Stokes system shows very fast convergence and is robust with respect to mesh sizes and reaction coefficients. Of course, the convergence is not satisfactory in the multi-domain case with cross points. But the number of needed iteration steps can be dramatically decreased by using an appropriate coarse space. A suitable choice of a coarse space for our new approach is subject of further research.

References

[1] V. Dolean, F. Nataf, and G. Rapin. Deriving a new domain decomposition method for the Stokes equations using the Smith factorization. Submitted, 2006.

[2] P. Le Tallec and A. Patra. Non-overlapping domain decomposition methods for adaptive *hp* approximations of the Stokes problem with discontinuous pressure fields. *Comput. Methods Appl. Mech. Engrg.*, 145:361–379, 1997.

[3] J. Li. A Dual-Primal FETI method for incompressible Stokes equations. *Numer. Math.*, 102:257–275, 2005.

[4] J. Li and O.B. Widlund. BDDC algorithms for incompressible Stokes equations. *SIAM J. Numer. Anal.*, 44:2432–2455, 2006.

[5] F. Nataf and G. Rapin. Construction of a new domain decomposition method for the Stokes equations. In *Domain Decomposition Methods in Science and Engineering XVI*, volume 55 of *Lecture Notes Comput. Sci. Engrg.*, pages 247–254. Springer, Berlin, 2007.

[6] L.F. Pavarino and O.B. Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. *Comm. Pure Appl. Math.*, 55:302–335, 2002.

[7] J.T. Wloka, B. Rowley, and B. Lawruk. *Boundary Value Problems for Elliptic Systems.* Cambridge University Press, Cambridge, 1995.

[8] X. Zhang. Domain decomposition algorithms for the biharmonic Dirichlet problem. In *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 119–126. SIAM, 1992.