Domain Decomposition Methods for the Stokes and Oseen equations using the Smith factorization

V. Dolean [1] F. Nataf [2] G. Rapin [3]

[1] **Laboratoire J.A. Diedonné**
Univ. de Nice Sophia-Antipolis, France

[2] **Laboratoire J.L. Lions**
Univ. Pierre et Marie Curie, France

[3] **Institute for Numerical and Applied Mathematics (NAM)**
Georg August University Göttingen, Germany

Paris
September 18, 2007

Outline

Smith Factorization (Smith, 1860)

Theorem

Let n be an integer and A be an invertible $n \times n$ matrix with polynomial entries $a_{ij}(\lambda)_{1\le i,j\le n}$ with resp. to λ.
$\Longrightarrow \quad \exists$ polynomial matrices E, D, F with

$$A = EDF$$

- $det(E)$, $det(F)$ are constants.
- D is a diagonal matrix.

Remarks:

- D is uniquely determined up to a reordering and multiplication of each entry by a constant.
- The inverses of E and F have also polynomial entries.

Computing the Smith factorization

D is uniquely defined by the formula defined as follows. Let $1 \leq k \leq n$,

- S_k is the set of all the submatrices of order $k \times k$ extracted from A.
- $Det_k = \{\mathrm{Det}(B_k) \backslash B_k \in S_k\}$
- LD_k is the largest common divisor of the set of polynomials Det_k.

Then,

$$D_{kk}(\lambda) = \frac{LD_k(\lambda)}{LD_{k-1}(\lambda)}, \quad 1 \leq k \leq n \tag{1}$$

(by convention, $LD_0 = 1$). In practice, the factorization can be computed “by hand” similarly to a Gauss factorization OR one can use the Maple routine called `Smith`.

How to Use the Smith factorization

Suppose $\mathcal{A}(\partial_x, \partial_y)$ is a partial differential operator and we need to solve the following system of PDEs:

$$\mathcal{A}(U) = b$$

The Fourier transform with respect to y, $\hat{\mathcal{A}}(\partial_x, k)$ is a polynomial matrix wrt to ∂_x. Let $\hat{\mathcal{A}} = EDF$.
Let $V = F(U)$, then it remains to solve the uncoupled scalar equations:

$$D(V) = E^{-1}b$$

The Smith factorization provides the possibility to analyze different aspects of the resolution of the PDEs by reducing them to equivalent scalar systems:
Preconditionning aspects of domain decomposition methods

Stokes Equations

$$\begin{aligned} -\nu\triangle \boldsymbol{u} + \nabla p + c\boldsymbol{u} &= \boldsymbol{f} \quad \text{in } \Omega \\ \nabla \cdot \boldsymbol{u} &= 0 \quad \text{in } \Omega \end{aligned}$$

- Simple model for incompressible flows
- Domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$
- Source term $\boldsymbol{f} \in [L^2(\Omega)]^d$, viscosity $\nu > 0$, reaction $c \geq 0$
- Stokes operator $\mathcal{S}_d(\boldsymbol{v}, q) := (-\nu\triangle \boldsymbol{v} + c\boldsymbol{v} + \nabla q, \nabla \cdot \boldsymbol{v})$

Existing Algorithms and Exactness

Existing Algorithms for the Stokes Equations

Neumann-Neumann type	Ainsworth, Sherwin ('99) Le Tallec, Patra ('97) Pavarino, Widlund ('02)
FETI	Li ('05)
BDDC	Li, Widlund ('06)
others	Quarteroni ('89), Bramble, Pasciak ('90)

Problem:
In opposite to the scalar case all these methods are not *exact* in the case of two subdomains consisting of the two half planes.
A method is called *exact*, if the preconditioned operator simplifies to the identity.

Main Idea

- Neumann-Neumann preconditioners are exact for many scalar equations like Laplace or Helmholz equations. (cf. ACHDOU ET AL. ('00) for the advection-diffusion equations)
- We use the Smith Factorization as a general tool to reduce the system to a set of uncoupled scalar equations.
- Starting with an exact algorithm for the corresponding scalar problems we derive a method for the Stokes equations which preserves this property.
- Same procedure can be applied to the Oseen eqations.

Application to the 2D Stokes Equations

- Consider the whole plain: $\Omega = \mathbb{R}^2$
- Fourier transform in y-direction (vertical) with dual variable k

$\Longrightarrow$ Stokes equations are equivalent to

$$\hat{\mathcal{S}}_2(\hat{\boldsymbol{u}}, \hat{p}) = \hat{\boldsymbol{g}}$$

with $\hat{\boldsymbol{u}} = (\hat{u}, \hat{v})$, $\hat{\boldsymbol{g}} = (\hat{f}_1, \hat{f}_2, 0)^T$ and

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

Idea: Interpret $\hat{\mathcal{S}}_2$ as matrix with polynomial entries in ∂_x

Smith Fact. for the 2D Stokes Equations

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

with

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

$$\hat{E}_2 = \hat{T_2}^{-1} \begin{pmatrix} ik\hat{\mathcal{L}}_2 & \nu\partial_{xxx} & -\nu\partial_x \\ 0 & \hat{T}_2 & 0 \\ ik\partial_x & -\partial_{xx} & 1 \end{pmatrix}$$

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

Reformulation of the Stokes Problem

- Let $(\hat{\boldsymbol{w}}, \hat{p})$ satisfy the Stokes equations

$$\hat{\mathcal{S}}_2(\hat{\boldsymbol{w}}, \hat{p}) = \hat{E}_2 \hat{D}_2 \hat{F}_2(\hat{\boldsymbol{w}}, \hat{p}) = \hat{\boldsymbol{g}} \quad \text{in } \mathbb{R}^2.$$

- Multiplying with $\hat{E}_2^{\,-1}$ yields

$$\hat{D}_2 \hat{F}_2(\hat{\boldsymbol{w}}, \hat{p}) = \hat{E}_2^{\,-1} \hat{\boldsymbol{g}} \quad \text{in } \mathbb{R}^2.$$

- Defining $\hat{\boldsymbol{u}} := \hat{F}_2(\hat{\boldsymbol{w}}, \hat{p})$ we obtain

$$\begin{array}{rcl} \hat{u}_1 & = & (E_2^{-1}\hat{\boldsymbol{g}})_1 \\ \hat{u}_2 & = & (E_2^{-1}\hat{\boldsymbol{g}})_2 \\ (\partial_{xx} - k^2)\hat{\mathcal{L}}_2 \hat{u}_3 & = & (E_2^{-1}\hat{\boldsymbol{g}})_3 \end{array} .$$

- Using $\hat{u}_3 = (\hat{F}_2(\hat{\boldsymbol{w}}, \hat{p}))_3 = \hat{w}_2$ and the inverse Fourier transform $\mathcal{F}_y^{-1}$ we get

$$\triangle \mathcal{L}_2 w_2 = \mathcal{F}_y^{-1}(\hat{E}_2^{-1} \hat{\boldsymbol{g}}_3).$$

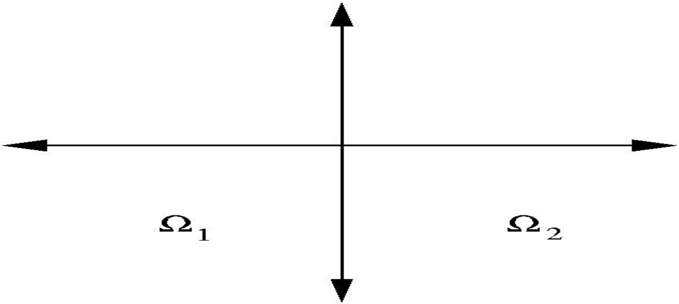
Remarks

- Multiplying with $\hat{E}_2^{-1}$ corresponds to a differentiation in x-direction.
- The Stokes problem can be mainly characterized by the fourth-order operator $\triangle(-\nu\triangle + c)$.
- The Stream function formulation yields the same differential operator in the 2D case.

Main Idea for Deriving DD Methods

- Deriving an efficient dd method for the scalar fourth-order problem.
- We consider a special geometry and express the domain decomposition method in terms of the Stokes problem.
- With the help of the Stokes equations the higher order interface conditions can be rewritten as lower order conditions.
- As a result we obtain a dd method for the 2D Stokes equations for this geometry.
- Generalize this algorithm to arbitrary domains.

Special Geometry



- $\Omega = \mathbb{R}^2$
- $\Omega_1 = \{(x, y) \in \mathbb{R}^2 \mid x < 0\ \}$
- $\Omega_2 = \{(x, y) \in \mathbb{R}^2 \mid x > 0\ \}$
- $\Gamma = \{(x, y) \in \mathbb{R}^2 \mid x = 0\ \}$

Efficient algorithm for the scalar problem

- Initial guess with

$$\mathcal{L}_2 u_2^{1,0} = \mathcal{L}_2 u_2^{2,0}, \quad u_2^{1,0} = u_2^{2,0} \text{ on } \Gamma$$

- Correction step ($i = 1, 2$)

$$\begin{aligned}
\triangle \mathcal{L}_2 v_2^{i,n} &= 0 \quad \text{in } \Omega_i \\
\frac{\partial}{\partial \boldsymbol{n}_i} \mathcal{L}_2 v_2^{i,n} &= -\frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{n}_1}\mathcal{L}_2 u_2^{1,n-1} + \frac{\partial}{\partial \boldsymbol{n}_2}\mathcal{L}_2 u_2^{2,n-1}\right) \text{ on } \Gamma. \\
\nu \frac{\partial v_2^{i,n}}{\partial \boldsymbol{n}_i} &= -\frac{1}{2}\nu\left(\frac{\partial u_2^{1,n-1}}{\partial \boldsymbol{n}_1} + \frac{\partial u_2^{2,n-1}}{\partial \boldsymbol{n}_2}\right) \text{ on } \Gamma.
\end{aligned}$$

- Update step ($i = 1, 2$)

$$\begin{aligned}
\triangle \mathcal{L}_2 u_2^{i,n} &= \mathcal{F}_y^{-1}(\hat{E}_2^{-1}\hat{\boldsymbol{g}}_3) \quad \text{in } \Omega_i \\
\mathcal{L}_2 u_2^{i,n} &= \mathcal{L}_2 u_2^{1,n-1} + \frac{1}{2}\left(\mathcal{L}_2 v_2^{1,n} + \mathcal{L}_2 v_2^{2,n}\right) \quad \text{on } \Gamma \\
u_2^{i,n} &= u_2^{i,n-1} + \frac{1}{2}(v_2^{1,n} + v_2^{2,n}) \quad \text{on } \Gamma.
\end{aligned}$$

Convergence

Theorem

Let $\Omega = \mathbb{R}^2$ *be decomposed into*
$\Omega_1 = \{(x, y) \in \mathbb{R}^2 \mid x < 0\}$, $\Omega_2 = \{(x, y) \in \mathbb{R}^2 \mid x > 0\}$.

- *The scalar algorithm converges in at most two steps.*

Remarks:

- Very natural interface conditions
- For the model case the algorithm possesses perfect convergence properties.
- The domain decomposition method of the Stokes equations will inherit these properties.

Next Steps

1. Rewrite the algorithm in terms of the Stokes equations (for the special geometry), use for example $\partial_x u_1 = -\partial_y u_2$ for the velocity (u_1, u_2).
2. Generalize it to arbitrary decompositions

Arbitrary decomposition

- Non-overlapping decomposition $\{\Omega_i\}_{i=1}^N$ of Ω, i.e.

$$\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega_i}, \qquad \Omega_i \cap \Omega_j = \emptyset, i \neq j$$

- Interface $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$, $\Gamma = \bigcup \Gamma_{ij}$
- Stress on the interface

$$\boldsymbol{\sigma}(\boldsymbol{u}, p) := \nu \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{n}} - p\boldsymbol{n}$$

- We use the notation $\boldsymbol{u_n}$ for the normal and $\boldsymbol{u_\tau}$ for the tangential part of the velocity $\boldsymbol{u}$. We also split the stress $\boldsymbol{\sigma}$ in $\boldsymbol{\sigma_n}$ and $\boldsymbol{\sigma_\tau}$.

Equivalent Algorithm for Stokes

- Initial guess $((\boldsymbol{u}_i^0, p_i^0))_{i=0}^N$ with

$$u^0_{i,\boldsymbol{\tau}_i} = u^0_{j,\boldsymbol{\tau}_j}, \quad \sigma_{\boldsymbol{n}_i}(\boldsymbol{u}_i^0, p_i^0) = -\sigma_{\boldsymbol{n}_j}(\boldsymbol{u}_j^0, p_j^0) \quad \text{on } \Gamma_{ij}$$

- Correction step

$$\left\{\begin{array}{l} \mathcal{S}_2(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) = 0 \quad \text{in } \Omega_i \\ \tilde{u}^{n+1}_{i,\boldsymbol{n}_i} = -\dfrac{1}{2}(u^n_{i,\boldsymbol{n}_i} + u^n_{j,\boldsymbol{n}_j}) \quad \text{on } \Gamma_{ij} \\ \sigma_{\boldsymbol{\tau}_i}(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) \\ \qquad = -\dfrac{1}{2}(\sigma_{\boldsymbol{\tau}_i}(\tilde{\boldsymbol{u}}_i^n, \tilde{p}_i^n) + \sigma_{\boldsymbol{\tau}_j}(\tilde{\boldsymbol{u}}_j^n, \tilde{p}_j^n)) \quad \text{on } \Gamma_{ij} \end{array}\right.$$

Equivalent Algorithm for Stokes

- Update step

$$\begin{cases} \mathcal{S}_2(\boldsymbol{u}_i^{n+1}, p_i^{n+1}) = \boldsymbol{f} \quad \text{in } \Omega_i \\ u_{i,\boldsymbol{\tau}_i}^{n+1} = u_{i,\boldsymbol{\tau}_i}^{n} + \frac{1}{2}(\tilde{u}_{i,\boldsymbol{\tau}_i}^{n+1} + \tilde{u}_{j,\boldsymbol{\tau}_i}^{n+1}) \quad \text{on } \Gamma_{ij} \\ \sigma_{\boldsymbol{n}_i}(\boldsymbol{u}_i^{n+1}, p_i^{n+1}) = \sigma_{\boldsymbol{n}_i}(\boldsymbol{u}_i^{n}, p_i^{n}) \\ \qquad + \frac{1}{2}(\sigma_{\boldsymbol{n}_i}(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) - \sigma_{\boldsymbol{n}_j}(\tilde{\boldsymbol{u}}_j^{n+1}, \tilde{p}_j^{n+1})) \text{ on } \Gamma_{ij}. \end{cases}$$

Remarks:

- The algorithm is very similar to the Neumann-Neumann method.
- In the case of $\Omega_1 = \{(x, y) \in \mathbb{R}^2 \mid x < 0\}$, $\Omega_2 = \{(x, y) \in \mathbb{R}^2 \mid x > 0\}$ we obtain convergence in two steps.

Extension to the Stokes Equations in 3D

- Fourier transform (with dual variables k and η)

$$\hat{\mathcal{S}}_3 = \begin{pmatrix} \hat{\mathcal{L}}_3 & 0 & 0 & \partial_x \\ 0 & \hat{\mathcal{L}}_3 & 0 & ik \\ 0 & 0 & \hat{\mathcal{L}}_3 & i\eta \\ \partial_x & ik & i\eta & 0 \end{pmatrix}$$

  where $\hat{\mathcal{L}}_3 := \nu(-\partial_{xx} + k^2 + \eta^2) + c$ is the Fourier transform of $\mathcal{L}_3 := -\nu\Delta + c$.
- Diagonal matrix of the Smith factorization

$$\hat{D}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \hat{\mathcal{L}}_3 & 0 \\ 0 & 0 & 0 & (\partial_{xx} - k^2 - \eta^2)\hat{\mathcal{L}}_3 \end{pmatrix}$$

- Thus the 3D-Stokes problem is determined by $\mathcal{L}_3$ and $\triangle\mathcal{L}_3$
- After similar computations we obtain exactly the same algorithm.

Extension to the Oseen Equations in 2D

Oseen equations
(Linearized Navier-Stokes equations)

$$\left\{\begin{array}{lcll} -\nu\Delta \boldsymbol{u} + \boldsymbol{b}\cdot\nabla\boldsymbol{u} + c\boldsymbol{u} + \nabla p & = & \boldsymbol{f} & \text{in } \Omega \\ \nabla\cdot\boldsymbol{u} & = & 0 & \text{in } \Omega. \end{array}\right.$$

- Oseen operator

$$\mathcal{O}_2(\boldsymbol{u}, p) = (-\nu\Delta\boldsymbol{u} + \boldsymbol{b}\cdot\nabla\boldsymbol{u} + c\boldsymbol{u} + \nabla p, \nabla\cdot\boldsymbol{u})^T$$

- Diagonal matrix of the Smith Factorization is the Fourier transform of

$$D_2^{\mathcal{O}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathcal{L}_2^{\mathcal{O}}\Delta \end{pmatrix}$$

with $\mathcal{L}_2^{\mathcal{O}} u = -\nu\Delta u + \boldsymbol{b}\cdot\nabla u + cu.$

Algorithm for the 2D Oseen equations

- Correction step

$$\begin{cases} \mathcal{O}_2(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) = 0 \quad \text{in } \Omega_i \\ \sigma_{\boldsymbol{\tau}_i}(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) - \frac{1}{2}(\boldsymbol{b}\cdot\boldsymbol{n}_i)\tilde{\boldsymbol{u}}_{i,\tau_i}^{n+1} = \\ \quad -\frac{1}{2}(\sigma_{\boldsymbol{\tau}_i}(\boldsymbol{u}_i^n, p_i^n) + \sigma_{\boldsymbol{\tau}_j}(\boldsymbol{u}_j^n, p_j^n)) \quad \text{on } \Gamma_{ij} \\ (-\nu\partial_{\boldsymbol{\tau}_i\boldsymbol{\tau}_i} + (\boldsymbol{b}\cdot\boldsymbol{\tau}_i)\partial_{\boldsymbol{\tau}_i} + c)\tilde{u}_{i,\boldsymbol{n}_i}^{n+1} - \frac{1}{2}(\boldsymbol{b}\cdot\boldsymbol{n}_i)\partial_{\boldsymbol{\tau}_i}\tilde{u}_{i,\boldsymbol{\tau}_i}^{n+1} = \gamma_{ij}^n, \Gamma_{ij} \end{cases}$$

with $\gamma_{ij}^n := -\frac{1}{2}(-\nu\partial_{\boldsymbol{\tau}_i\boldsymbol{\tau}_i} + (\boldsymbol{b}\cdot\boldsymbol{\tau}_i)\partial_{\boldsymbol{\tau}_i} + c)\left(u_{i,\boldsymbol{n}_i}^n + u_{j,\boldsymbol{n}_j}^n\right)$.

- Update step

$$\begin{cases} \mathcal{O}_2(\boldsymbol{u}_i^{n+1}, p_i^{n+1}) = \boldsymbol{f} \quad \text{in } \Omega_i \\ u_{i,\boldsymbol{\tau}_i}^{n+1} = u_{i,\boldsymbol{\tau}_i}^n + \frac{1}{2}(\tilde{u}_{i,\boldsymbol{\tau}_i}^{n+1} + \tilde{u}_{j,\boldsymbol{\tau}_j}^{n+1}) \quad \text{on } \Gamma_{ij} \\ \sigma_{\boldsymbol{n}_i}(\boldsymbol{u}_i^{n+1}, p_i^{n+1}) = \sigma_{\boldsymbol{n}_i}(\boldsymbol{u}_i^n, p_i^n) + \delta_{ij}^{n+1} \text{ on } \Gamma_{ij} \end{cases}$$

with $\delta_{ij}^{n+1} = \frac{1}{2}(\sigma_{\boldsymbol{n}_i}(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) - \sigma_{\boldsymbol{n}_j}(\tilde{\boldsymbol{u}}_j^{n+1}, \tilde{p}_j^{n+1}))$.

Numerical Tests

Consider a rectangle $\Omega := (0,4) \times (0,1)$:

$$\begin{aligned} -\nu \triangle \boldsymbol{u} + c\boldsymbol{u} + \nabla p &= \boldsymbol{f} \text{ in } \Omega \\ \nabla \cdot \boldsymbol{u} &= 0 \text{ in } \Omega \end{aligned}$$

and suitable boundary conditions for $\nu = 1$, $c = 10^{-5}, 10^{0}, 10^{2}$.

Reference Solution:

$$\begin{aligned} \boldsymbol{u}(x,y) &= \begin{pmatrix} \sin^3(\pi x)\sin^2(\pi y)\cos(\pi y) \\ -\sin^2(\pi x)\sin^3(\pi y)\cos(\pi x) \end{pmatrix}, \\ p &= x^2 + y^2 \end{aligned}$$

Discretization:
Finite Volume discretization with staggered grids and pressure stabilization, different mesh sizes h.

Two-subdomain case

Different reaction

regular decomposition: 2×1 subdomains
mesh size: $h = 1/96$
Stopping criterion: Reduction of the error by 10^{-6}

c	new_{it}	nn_{it}	new_{GMRES}	nn_{GMRES}
10^2	2	15	1	6
1	2	15	1	6
10^{-3}	2	15	1	6
10^{-5}	2	15	1	6

Two-subdomain case

Different mesh sizes

regular decomposition: 2×1 subdomains
reaction: $c = 10^{-5}$
Stopping criterion: Reduction of the error by 10^{-6}

h	new_{it}	nn_{it}	new_{GMRES}	nn_{GMRES}
$1/24$	2	14	1	6
$1/48$	2	15	1	6
$1/96$	2	15	1	6

Stripwise decomposition - regular case

regular decomposition: $N \times 1$ subdomains
mesh size: $h = 1/96$
Stopping criterion: Reduction of the error by 10^{-6}

reaction $c = 10^{-5}$:

N	new_{it}	nn_{it}	new_{GMRES}	nn_{GMRES}
2	2	15	1	6
4	-	-	8	-
6	-	-	15	-
8	-	-	21	-

reaction $c = 10^{2}$:

N	new_{it}	nn_{it}	new_{GMRES}	nn_{GMRES}
2	2	15	1	6
4	35	-	5	9
6	-	-	7	15
8	-	-	10	21

Stripwise decomposition - non-regular case

decomposition: 4×1 subdomains
width of subdomain Ω_i**:** l_i
mesh size: $h = 1/96$
Stopping criterion: Reduction of the error by 10^{-6}

c	N	it_{New}	it_{NN}	ac_{New}	ac_{NN}
10^{-5}	[16, 32, 16, 32]	-	-	9	-
	[16, 48, 16, 16]	-	-	10	-
	[48, 16, 16, 16]	-	-	12	-
10^{0}	[16, 32, 16, 32]	-	-	8	14
	[16, 48, 16, 16]	-	-	10	13
	[48, 16, 16, 16]	-	-	12	17
10^{2}	[16, 32, 16, 32]	74	-	5	12
	[16, 48, 16, 16]	-	-	6	11
	[48, 16, 16, 16]	-	-	6	14

General case

regular decomposition: $N \times N$ subdomains
mesh size: $h = 1/96$
Stopping criterion: Reduction of the error by 10^{-6}

c	$N \times N$	it_{New}	it_{NN}	ac_{New}	ac_{NN}
10^{-5}	2x2	-	-	9	13
	3x3	-	-	28	-
	4x4	-	-	40	-
10^{0}	2x2	-	-	9	13
	3x3	-	-	30	28
	4x4	-	-	39	39
10^{2}	2x2	61	-	7	11
	3x3	-	-	22	21
	4x4	-	-	27	27

Summary and Outlook

Summary

- Introduction of a new domain decomposition for the 2D and 3D Stokes problem.
- We could prove perfect convergence for a model problem.
- Theoretical results could be validated numerically.
- Extension to the Oseen case. Convergence of the algorithm is theoretically independent of the Reynolds number.

Outlook

- Analyzing the general case.
- Introduction of suitable coarse spaces.
- Analyzing and performing numerical tests for the Oseen equations.