Balancing Domain Decomposition Preconditioners for Non-symmetric Problems

Adviser: Marcus Vinicius Sarkis
Student: Duilio Tadeu da Conceição Junior

May 9, 2006

Instituto Nacional de Matemática Pura e Aplicada

Agência Nacional do Petróleo
PRH-32

Balancing Domain Decomposition Preconditioners for Non-symmetric Problems

Author: Duilio Tadeu da Conceição Jr.

Adviser: Marcus Sarkis

Rio de Janeiro
May 9, 2006

To my parents Duilio and Regina
To my brother William

Abstract

The core-flow is a technique to make the production and the transportation of heavy oil through pipe technically and economically viable. In this technique we form a film of water around the heavy oil core in the pipe. The numerical simulation is of great importance for the development of the technology. However, the software for numerical simulation available nowadays are inefficient when dealing with problems that have high jumps in the viscosity such as the core-flow model. In order to obtain a robust and efficient preconditioner for such model we focus our attention on the linearized Navier-Stokes equations and on the advection-diffusion equation.

Balancing domain decomposition (BDD) algorithms are substructuring methods that have been previously shown to be effective for solving large finite element approximations of symmetric positive definite elliptic and certain saddle point problems with jump in the coefficients. In this thesis, we extend the BDD methods for solving the stabilized advection-diffusion equation, the enhanced Stokes equations and the stabilized Oseen equations on unstructured meshes. Our methods are shown to be almost scalable with respect to the number of subdomains or processors. Our work is an extension of the previous works by Mandel, Pavarino, Widlund and Goldfeld on BDD methods for Poisson, Stokes and elasticity equations, and an extension of the BDDC method of Dohrmann for Poisson equation.

We consider a Galerkin least square stabilization on the discretization of the model equations, since when the advective part dominates the diffusive one, the numerical solution presents spurious oscillations when the mesh is not refined enough to resolve the small scales. The discretization of these models lead to non-symmetric linear systems, which introduces some difficulties on the design of effective preconditioners. The linear system has positive definite symmetric part, however very small compared to the skew-symmetric part. The resulting linear system is very ill-conditioned and presents a real challenge for preconditioning purposes.

In this thesis, we develop substructuring methods for non-symmetric problems using non-overlapping domain decompositions where we partition the unknowns into interior and interface variables. We then perform the elimination of the interior degrees of freedom by static condensation obtaining a Schur complement system. We develop hybrid preconditioners of BDD type for the Schur complement systems

obtained from the stabilized advection-diffusion equation, the Stokes equations and the Oseen equations. The coarse space is based on the kernel of the operators in order to obtain solvability of the Neumann local problems and to obtain scalability by broadcasting the solution among the subdomains. In addition, in the case of Stokes equations the coarse space should be rich enough to obtain inf-sup stability on the coarse space.

We also develop coarse spaces suitable for the implementation on unstructured meshes of the Stokes/Oseen version of the preconditioner. We present numerical results for the implemented coarse spaces. All the algorithms discussed have been implemented in parallel codes for unstructured meshes using the PETSc library. Those implementations have been successfully tested on large sample problems.

Keywords: Domain decomposition, balancing preconditioner, non-symmetric problems, advection-diffusion, Stokes, Oseen, BDD, BDDC.

Agradecimentos

Ao meu orientador Marcus Sarkis pela orientação, estímulo e persistência, estando sempre disponível para discutir e cheio de novas idéias.

Pela grande ajuda com a biblioteca PETSc e paciência, e por ceder parte do código paralelo gostaria de agradecer ao professor Paulo Goldfeld.

Pelas frutíferas discussões, idéias e paciência agradeço ao professor Tarek Mathew.

A meus grandes amigos Juan, Henrique, Wanderson Lambert e Christian pela grande ajuda durante o desenvolvimento da tese.

Aos professores Dan Marchesin e André Nachbin que sempre estiveram disponíveis para ajudar.

Aos funcionários do laboratório Letícia, Beata e Sergio pela prestatividade e acima de tudo amizade.

Ao funcionário Daniel Albuquerque pela enorme prestatividade e disposição em ajudar na programação C e com a biblioteca PETSc.

Ao professor Dinamérico pelos excelentes conselhos, motivação e grande estímulo.

Aos professores Nilson e Jorge Delgado pelo estímulo que deram nestes anos de doutorado.

Aos grandes amigos Wanderson Bispo, Flávio, Leandro, Sandra e Julia que sempre deram força e motivação.

A Amaury, Daniel Alfaro, Monique e Adan pelo estimulo nos mais diversos momentos.

Também não poderia deixar de agradecer a Sofy, Susan, Ana Luz, Grigori Chapiro (meu grande fã), Almir, Jhon Mira (meu segundo maior fã), Freddy, Javier, Damián, Airon, Vanessa Barros, Dália e tantas outras ótimas amizades que fiz ao longo destes anos no IMPA

Aos amigos da UFF que fiz durante meu mestrado Angélica, Marcelo, Maryelen, Daniel, Michel e Carolina que mesmo distantes sempre estiveram torcendo.

Aos amigos do LNCC, em especial Maicon e Julio.

A toda minha família que sempre me apoiou.

Gostaria de agradecer a ANP que através do PRH-32 foi responsável pelos meus quatro anos de suporte financeiro.

Duilio Tadeu da Conceição Junior
Rio de Janeiro, Brasil.
May 9, 2006.

Contents

List of Tables

List of Figures

CHAPTER 1

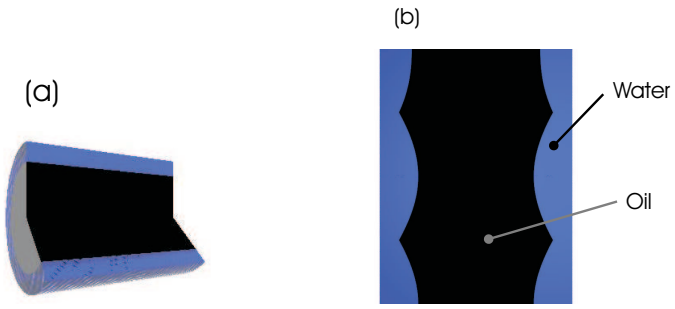
Introduction

The transportation of fluids of very high viscosity (such as heavy oil in petroleum industry) through pipe is very inefficient in terms of energy consumption and it is, in general, technically limited to be implemented. In petroleum industry the production and the transportation through pipe of heavy oil lead to analougous difficulties. One of the most prominent technology in petroleum engineering to make the production and the transportation of heavy oil through pipe technically and economicaly viable is the *core-flow* technique. It is generated a lubricated flow in the pipe using water in the annular space between the pipe wall and the the oil core (see Figure 1.1(a)). The water being immiscible in the oil acts as a lubricant for the oil core. By other hand, as a gift of nature, since the oil is very viscous the core tends to keep away of the pipe wall.

This technique was first discussed in Issacs and Speed (1904, US Patent No. 759374) for the water lubrication of lighter oils, although this technique is more effective when the oil is very viscous. Since then oil companies have had an intermittent interest in the technology. However, the technology seems to have never been tested for heavy and ultraviscous oil production/transportation. This technology could be a very good alternative in deep-water fields such as in the Brazilian coast, since no heat addition is required.

An introduction to the model equations can be found in [29] and also some stability analysis of the interface using assimptotic analysis. A review on the research in core-flow is presented in [28]. Lately in

Figure 1.1: (a) 3-D visualization of the perfect core-flow in a pipe section and (b) section sketching the bamboo waves in the interface between the heavy oil and the water.



Brazil, the experimental research of this technology have been carried out by Bannwart et al. in [4, 5]. In [2, 35], it is presented numerical experiments for the analysis of the interface's stability between the fluids.

In laboratory experiments it is verified that the pressure gradient necessary to transport the heavy oil using the core-flow technique is of the same order for transporting only water, which shows the great economic advantage of core-flow. These experiments also show that when the oil core is sufficiently fast so that the core is continuous, the interface shows the formation of bamboo waves, as sketched in Figure 1.1(b).

The numerical simulation is of great importance for the development of the technology. However, the software for numerical simulation available nowadays are inefficient when dealing with problems that have high jumps in the viscosity such as the core-flow model. The system of equations modelling the core-flow is composed of two Navier-Stokes equations (one for each phase) and an interface equation relating the jump of the normal stress and the jump in pressure accross the interface, and the mean curvature of the surface (i.e., the interface).

Observing that the numerical solution of the Navier-Stokes equations can be made by discretizing in space using finite element method (FEM, [27, 6]) and in time using Newton's method. This leads essentially to a problem of solving the linearized Navier-Stokes equations, also known as Oseen equations. Hence, the main difficulty to obtain an efficient preconditioner for the core-flow model is the development of a preconditioner for the Oseen equations problems with high jump

in the viscosity.

Balancing domain decomposition (BDD) preconditioners have been extensivelly used to solve Stokes problems with high jump in viscosities with great efficiency. The extension of these type of preconditioners to the Oseen problems is of great interest for the numerical simulation of the Navier-Stokes equations. In order to develop such preconditioners, we first develop BDD preconditioners for the steady advection-difusion equation which can be thought as the scalar version of the Oseen equations. The steady advection-difusion equation is the Poisson equation with the addition of an advective term $\boldsymbol{a}.\nabla u$. The advection-diffusion equation describe the transport of solutes in groundwater and surface water, the movement of aerosol and trace gases in the atmosphere, and many other important applications.

The goal of domain decomposition methods is to solve efficiently on parallel machines the numerical solution of partial differential equations that model a physical problem, such as incompressible fluid flow. The most desirable property of a domain decomposition preconditioner is scalability with respect to the number of subdomains, which can be defined in the following way: suppose that to solve a linear system of size n on N processors and the preconditioined iterative method takes p iterations to converge, then a domain decomposition preconditioner is said to be *scalable* if solving the new linear system of size $2n$ on $2N$ processors, the iterative method takes p iterations to converge.

The most common domain decomposition methods are additive and hybrid Schwarz methods, and iterative substructuring methods ([47, 52]). We restrict our development to finite element discretizations ([27, 6]), however Schwarz methods and others domain decomposition methods could be applied to other kind of discretizations such as finite differences.

The Schwarz method is an overlapping method, i.e., the domain is decomposed into overlapping subregions with an overlap of size δ. This preconditioner uses the local stiffness matrices of the subdomains to build an additive preconditioner. The convergence of Schwarz methods depend on the overlapping size δ.

In substructuring methods, the domain is partitioned into non-overlapping subdomains and the variables of the linear system which discretize a PDE are splitted into interior variables and interface variables. Using static condensation, the interior variables are eliminated obtaining a Schur complement system in the interface variables. The main methods of this type are the FETI (Finite Element Tearing and Interconnecting [18, 17, 32]), FETI-DP [16, 33, 34], BDD (Balancing

Domain Decomposition) and the BDDC (Balancing Domain Decomposition with Constraints). However these preconditioners are very efficient they are developed and built based on the structure of the discrete model of the PDE.

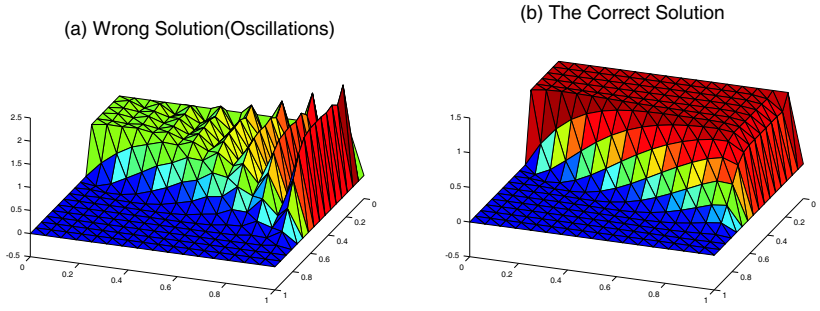
The BDD and BDDC methods, which are the focus of this thesis, have been tested successfully on several challenging large scale applications [43, 23, 42, 39, 12]. Coarse space and weighting diagonal matrices play crucial roles in making the BDD algorithms scalable with respect to the number of subdomains guaranteeing the consistency of the local Neumann subproblems on each iteration of the preconditioned system. The first BDD method introduced was due to Le Tallec and De Roeck [43] for the Poisson equation and without a coarse space, thus obtaining a non scalable preconditioner with inconsistent local problems. Later, in the work of Mandel [39], a coarse space was introduced based on the kernel of the Laplace operator thus solving the unresolved difficulties of the early version. Extensions of the BDD preconditioner for elliptic problems with possibly large jumps on coefficients were treated subsequently in [40, 14, 45, 46]. The extension of the BDD preconditioner for the Stokes equations was made only recently in the work by Pavarino and Widlund [42]. For the Stokes problem not only are the local Neumann problems singular for floating subdomains but additionally the boundary values of the local Dirichlet problems should satisfy the zero flux condition on the boundary of the subregions. Such issues are discussed in detail in [42] and in Chapter 3.

The BDDC method were introduced by C. Dohrmann in [12] for the Poisson equation and its convergence analysis in [41] shows similar bounds as for the BDD preconditioning. This preconditioner shares some similarities with the BDD method and the FETI-DP method. In [37], Li and Widlund extend the BDDC method for the Stokes case and its numerical results show that the convergence of the BDDC is similar to the FETI-DP method.

An analysis of some saddle-point preconditioners for the discrete Oseen equations is presented in [15].

Preconditioning techniques for non-symmetric problems has been proposed under many domain decomposition point of view, although a research under the light of balancing domain decomposition is missing in the literature. In [31], Kay, Loghin and Wathen consider a preconditioner based on the Green's tensor for Oseen equations that has a little dependence on the mesh size. Overlapping methods have been considered in [10, 9, 11]. In [1], Achdou et al. developed a Neumann-

Figure 1.2: Plot of the discrete solution of the advection-diffusion equation on a mesh of size 16×16, viscosity=0.01 and advection field=(1,3). In picture (a) we plot the solution of the system without stabilization and in picture (b) we plot the solution of the system with stabilization.



Neumann type preconditioner, by using Robin boundary condition instead of a Neumann one. Toselli extended the FETI preconditioner in his work [51], obtaining iteration counts that are very similar to those for the corresponding Robin-Robin methods in [1]. In this thesis we develop coarse problems for balancing domain decomposition methods that are more efficient obtaining almost scalability.

The advection-diffusion equation and the Oseen equations present spurious oscillations (see Figure 1.2) when the mesh is not fine enough to resolve the boundary and interior layers. The introduction of a stabilization on the discrete variational formulation of the problem is essential and is introduced in our models. These stabilizations where introduced in [7, 8] by Brooks and Hughes as the stream-line upwind Petrov-Galerkin (SUPG). In [21], Hughes and Franca developed the Galerkin least square stabilization which is better than the SUPG for higher order discretizations. Subsequently, Douglas and Wang [13] introduced a modification in the GLS stabilization obtaining a stabilization less sensible to the choice of the stabilization parameters. A similar formulation is also proposed by Lube and Auge [38].

In Chapter 2 we present the stabilized advection-diffusion equation model. Our main concern in this chapter is the development of the BDD and BDDC methods in the advective regimes.

Chapter 3 is devoted to the Stokes equations and the development of coarse spaces for the BDD preconditioner suitable for unstructured meshes. We also develop higher order discretizations, which by using a static condensation, can be thought as a stabilized lower order

discretization.

In Chapter 4 the BDD method for Stokes equations is extended to the stabilized Oseen equations exploring the kernel of the Oseen equations.

CHAPTER 2

BDD Preconditioners for Advection-Diffusion Equation

The spaces $L^2(\Omega)$ and $H^1(\Omega) = \{\boldsymbol{v} \in L^2(\Omega); \quad (\nabla \boldsymbol{v}, \nabla \boldsymbol{v})_\Omega < \infty\}$ are Hilbert spaces in which we introduce the norms

$$\|u\|_0^2 = \int_\Omega u^2(x)\ \mathrm{dx}$$

and

$$\|u\|_1^2 = \|u\|_0^2 + |u|_1^2$$

where $|\cdot|_1$ is the semi-norm

$$|u|_1^2 = \int_\Omega \nabla u.\nabla u\ \mathrm{dx}.$$

We also define the space

$$H^1_{0,\gamma}(\Omega) = \{v \in H^1(\Omega);\ v|_\gamma = 0\},$$

where $\gamma \subset \partial\Omega$ is a subset with a positive measure. In the case we omit the set γ it should be considered as $\partial\Omega$.

2.1 The Advection-Diffusion Equation

We assume $\Omega \subset \mathbb{R}^d$ to be a polygonal domain. The model problem is

$$\begin{cases} -\nu\triangle u + \boldsymbol{a}.\nabla u &= f \quad \text{in } \Omega \\ u &= u_d \quad \text{on } \partial\Omega \end{cases} \tag{2.1}$$

where $\nu > 0$ is the diffusion coefficient and $\boldsymbol{a} \in L^\infty(\Omega)^d$ is the advection vector field which we assume to be incompressible, i.e., $\nabla.\boldsymbol{a} = 0$. Since the reduction of the non-null Dirichlet u_d to a null one does not present difficulties, then we assume for simplicity $u_d = 0$.

The variational formulation is defined on the space $V = H_0^1(\Omega)$.

The following equation provides the advective term of the variational formulation of the problem and can be verified using Green formula.

Lemma 1. *Given $\boldsymbol{a} \in L^\infty(\Omega)^d$ and $u,\ v \in H^1(\Omega)$*

$$\begin{aligned}(\boldsymbol{a}.\nabla u, v)_\Omega = \frac{1}{2}\big[(\boldsymbol{a}.\nabla u, v)_\Omega - (u, \boldsymbol{a}.\nabla v)_\Omega\big] \\ - \frac{1}{2}\int_\Omega (\nabla.\boldsymbol{a})uv\ dx - \frac{1}{2}\int_{\partial\Omega}(\boldsymbol{a}.\boldsymbol{n})uv\ dx\end{aligned} \tag{2.2}$$

where $\boldsymbol{n}$ is the outward normal to Ω.

Since we assume $\nabla.\boldsymbol{a} = 0$ and $v \in V$ the resulting formula of Lemma 1 simplifies, and then we obtain that the advective term of the variational formulation is skew-symmetric. Using the last equation we obtain the following variational formulation of the advection-diffusion equation

Find $u \in V$ such that

$$\tilde{a}(u, v) = \widetilde{F}(v) \quad \forall u \in V \tag{2.3}$$

where $\tilde{a}(u, v) = a_{\text{diff}}(u, v) + a_{\text{skew}}(u, v)$ in which

$$a_{\text{diff}}(u, v) = \nu \int_\Omega \nabla u.\nabla v\ \mathrm{dx}$$

and

$$a_{\text{skew}}(u, v) = \frac{1}{2}\big[(\boldsymbol{a}.\nabla u, v)_\Omega - (u, \boldsymbol{a}.\nabla v)_\Omega\big].$$

Furthermore, $\widetilde{F}(v) = (f, v)_\Omega$.

The decomposition of the bilinear form $\tilde{a}(\cdot,\cdot)$ in these two bilinear forms is very useful for analysis and for numerics since $a_{\text{diff}}(\cdot,\cdot)$ is symmetric and $a_{\text{skew}}(\cdot,\cdot)$ is skew-symmetric.

In the case of non-null advection field $\boldsymbol{a}$ the bilinear form $\tilde{a}(\cdot,\cdot)$ is not symmetric, however it is still coercive. In fact, using Friedrichs inequality, the continuous bilinear form $a_{\text{diff}}(\cdot,\cdot)$ is coercive. In addition, $a_{\text{skew}}(\cdot,\cdot)$ is skew-symmetric, so $\tilde{a}(\cdot,\cdot)$ is coercive in V. This implies, using the Lax-Milgram theorem, that the variational problem (2.3) has a unique solution in V.

2.2 Discretization

Let $\mathcal{T}_h$ be a regular triangulation of the domain Ω where h is the characteristic mesh size. We discretize the space V with continuous and piecewise linear functions on each element of the mesh, i.e,

$$V_h = \{v \in V;\ v|_K \in P_1(K)\ \forall K \in \mathcal{T}_h\}.$$

The discrete variational formulation of the problem then reads

Find $u \in V_h$ such that

$$\tilde{a}(u,v) = \widetilde{F}(v) \quad \forall v \in V_h. \tag{2.4}$$

This leads to a non-singular linear system $Au = b$. We note however that such Galerkin discretization is numerically unstable for advective dominated problems, hence we consider stabilized finite elements.

2.2.1 Stabilization of Advective Problems

Let us define the Péclet number as the quotient of inertia and viscous forces in the discrete model, i.e.,

$$\mathrm{Pe} = \frac{\|\boldsymbol{a}\|_\infty L}{\nu}, \tag{2.5}$$

where $\|\boldsymbol{a}\|_\infty = \sup_{x\in\Omega}\|\boldsymbol{a}(x)\|_2$ and L is the characteristic size of the domain. We have a *diffusive-dominated* problem when $\mathrm{Pe} \leq O(1)$, and an *advection-dominated* problem when $\mathrm{Pe} > O(1)$. The discrete Galerkin advection-dominated problem (2.4) is numerically unstable and the solution will usually present spurious oscillations when the mesh size is not fine enough.

In order to avoid such instabilities we introduce a stabilization in a Galerkin least square fashion in the variational formulation (2.3) of the problem (2.1). We present here the main stabilization formulations. The first stabilization of this kind to appear was the SUPG (streamline-upwind-Petrov-Galerkin) introduced by Hughes and Brooks (see [7, 8, 25, 26]). Later, the Galerkin least square stabilization was developed by Franca in [21] and subsequently the Douglas-Wang modification of the GLS method was introduced in [13].

The stabilized formulation is

$$a(u,v) = F(v)$$

where for each stabilization we have the following bilinear and linear forms:

1. SUPG:

$$a(u,v) = \tilde{a}(u,v) + \sum_{K\in\mathcal{T}_h} \tau_K(-\nu\triangle u + \boldsymbol{a}.\nabla u, \boldsymbol{a}.\nabla v)_{0,K}$$
$$F(v) = (f,v) + \sum_{K\in\mathcal{T}_h} \tau_K(f, \boldsymbol{a}.\nabla v)_{0,K}$$

2. GLS:

$$a(u,v) = \tilde{a}(u,v) + \sum_{K\in\mathcal{T}_h} \tau_K(-\nu\triangle u + \boldsymbol{a}.\nabla u, -\nu\triangle v + \boldsymbol{a}.\nabla v)_{0,K}$$
$$F(v) = (f,v) + \sum_{K\in\mathcal{T}_h} \tau_K(f, -\nu\triangle v + \boldsymbol{a}.\nabla v)_{0,K}$$

3. DW:

$$a(u,v) = \tilde{a}(u,v) + \sum_{K\in\mathcal{T}_h} \tau_K(-\nu\triangle u + \boldsymbol{a}.\nabla u, \nu\triangle v + \boldsymbol{a}.\nabla v)_{0,K}$$
$$F(v) = (f,v) + \sum_{K\in\mathcal{T}_h} \tau_K(f, \nu\triangle v + \boldsymbol{a}.\nabla v)_{0,K}$$

Remark 2. *When employing linear elements the Laplacian term in the summation above vanishes, then the three stabilization methods coincide reducing to the SUPG stabilization method.*

In this chapter, we write V for V_h. Once we discretize the space with piecewise linear elements, by the previous remark, we redefine the bilinear form $\tilde{a}(\cdot,\cdot)$ that discretizes the problem (2.1) to incorporate a stabilization as

$$a(u,v) = a_{\text{diff}}(u,v) + a_{\text{skew}}(u,v) + \sum_{K\in\mathcal{T}_h} \tau_K(\boldsymbol{a}.\nabla u, \boldsymbol{a}.\nabla v)_{0,K}$$

and consequently, the right hand side is

$$F(v) = (f,v) + \sum_{K\in\mathcal{T}_h} \tau_K(f, \boldsymbol{a}.\nabla v)_{0,K}$$

Before defining the local stabilization parameter τ_K, we introduce as in [20] the local Péclet number for any element K in $\mathcal{T}_h$ as

$$\text{Pe}_K = \frac{m_h \|\boldsymbol{a}\|_{\infty,K} h_K}{2\nu} \tag{2.6}$$

where $m_h = \min\{1/3, 2/C_h\}$ and C_h is the constant appearing in the following inverse inequality

$$\sum_{K\in\mathcal{T}_h} h_K^2 \|\triangle v\|_{0,K}^2 \leq C_h \|\nabla \boldsymbol{v}\|_{0,K}^2 \quad v \in V_h.$$

An analysis of the discretization space V shows that for P_1 elements the constant C_h is 0, since for a linear function $\triangle v = 0$ (see [24]).

The parameter of stabilization τ_K is defined on each element $K \in \mathcal{T}_h$ as

$$\tau_K = \frac{h}{2\|\boldsymbol{a}\|_{\infty,K}} \xi(\mathrm{Pe}_K)$$

where

$$\xi(\mathrm{Pe}_K) = \begin{cases} \mathrm{Pe}_K; & 0 \leq \mathrm{Pe}_K < 1 \\ 1; & \mathrm{Pe}_K \geq 1. \end{cases}$$

We also can decompose the bilinear form $a(u,v)$ as

$$a(u,v) = a_{\mathrm{symm}}(u,v) + a_{\mathrm{skew}}(u,v)$$

where

$$a_{\mathrm{symm}}(u,v) = \nu(\nabla u, \nabla v) + \sum_{K\in\mathcal{T}_h} \tau_K(\boldsymbol{a}.\nabla u, \boldsymbol{a}.\nabla v)_{0,K}.$$

We define the semi-norm

$$|v|_{\mathrm{symm}}^2 = \nu\|\nabla v\|_0^2 + \sum_K \tau_K(\boldsymbol{a}.\nabla v, \boldsymbol{a}.\nabla v)_{0,K}$$

By Poincaré inequality, the bilinear form $a(\cdot,\cdot)$ is coercive in the semi-norm $|v|_{\mathrm{symm}}$.

On each Ω_i we define the local bilinear form $a^{(i)}(\cdot,\cdot)$ by replacing Ω in the integrals that defines $a(\cdot,\cdot)$ by Ω_i. Similarly we define $a^{(i)}_{\mathrm{symm}}(\cdot,\cdot)$, $a^{(i)}_{\mathrm{skew}}(\cdot,\cdot)$ and the semi-norm $|\cdot|_{\mathrm{symm},\Omega_i}$.

2.3 BDD Preconditioning for Advection-Diffusion

The extension of the balancing preconditioner from Poisson equation to advection-diffusion equation follows naturally by formulating the problem in terms of symmetric and skew-symmetric parts, however

its analysis is quite difficult and interesting since the operator is not longer symmetric.

Let us decompose the domain Ω into N non-overlapping connected subdomains Ω_i and let $\Gamma = (\cup_{i=1}^{N} \partial\Omega_i) \backslash \partial\Omega$ be the interface between the subdomains, so that we have

$$\Omega = (\cup_i \Omega_i) \cup \Gamma.$$

Furthermore, we define $\Gamma_i = \partial\Omega_i \backslash \partial\Omega$, the interface associated to subdomain Ω_i. We also define the characteristic size of the subdomain Ω_i by H_i and let $H = \max_{i=1,\cdots,N} H_i$.

In order to perform the Schur complement of the linear system we introduce the reordering of the variable u as

$$\begin{array}{ll} u_I & \leftarrow \text{interior variables} \\ u_\Gamma & \leftarrow \text{variables on interface } \Gamma, \end{array}$$

so that the linear system $Au = f$ takes the following form

$$\begin{pmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} u_I \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} f_I \\ f_\Gamma \end{pmatrix}.$$

We eliminate the interior variable u_I by static condensation obtaining the Schur complement system

$$S u_\Gamma = \tilde{f}_\Gamma \tag{2.7}$$

where

$$S = A_{\Gamma\Gamma} - A_{\Gamma I} A_{II}^{-1} A_{I\Gamma} \tag{2.8}$$

and $\tilde{f}_\Gamma = f_\Gamma - A_{\Gamma I} A_{II}^{-1} f_I$. Once solved the linear system (2.7), we can recover the interior solution u_I by solving the linear system

$$u_I = A_{II}^{-1}(f_I - A_{I\Gamma} u_\Gamma)$$

which can be performed in parallel.

Given u defined on Γ_i, we define its *local advection-diffusion extension* as

$$\mathcal{H}_{\text{adv}}^{(i)} u = \begin{pmatrix} -A_{II}^{(i)^{-1}} A_{I\Gamma}^{(i)} u \\ u \end{pmatrix}$$

This is equivalently defined as the solution of the discrete variational problem on $\overline{\Omega_i}$ by

$$\begin{cases} a^{(i)}(\mathcal{H}_{\text{adv}}^{(i)} u, v) = 0 & \forall v \in H_0^1(\Omega_i) \cap V \\ \mathcal{H}_{\text{adv}}^{(i)} u = u & \text{on } \Gamma_i \\ \mathcal{H}_{\text{adv}}^{(i)} u = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \end{cases}$$

Using these local extensions, we define the advection-diffusion extension operator $\mathcal{H}_{\text{adv}}$ of a vector u defined on Γ as

$$\mathcal{H}_{\text{adv}}u = \begin{cases} \mathcal{H}^{(i)}_{\text{adv}}u & \text{in } \Omega_i \\ u & \text{on } \Gamma \\ 0 & \text{on } \partial\Omega \end{cases}$$

Equivalently, $\mathcal{H}_{\text{adv}}u = \begin{pmatrix} -A_{II}^{-1}A_{I\Gamma}u \\ u \end{pmatrix}$.

Since the bilinear forms $a^{(i)}(\cdot,\cdot)$ are coercive on $H^1_0(\Omega_i)\cap V$, by Lax-Milgram's theorem the harmonic extensions are well defined.

Reordering the interior variables by subdomain, we obtain that

$$A_{II} = \begin{pmatrix} A_{II}^{(1)} & & 0 \\ & \ddots & \\ 0 & \cdots & A_{II}^{(\mathrm{N})} \end{pmatrix},$$

which allow us to apply A_{II}^{-1} in parallel, that is, by solving the Dirichlet local problems in parallel. We remark that this is the most expensive part when applying the Schur complement S to a vector $u \in V_\Gamma$.

If we define the local Schur complements $S_i = A^{(i)}_{\Gamma\Gamma} - A^{(i)}_{\Gamma I}{A^{(i)}_{II}}^{-1}A^{(i)}_{I\Gamma}$ and the restriction operator $\overline{R}_i : \Gamma \to \Gamma_i$, one can easily check that

$$S = \sum_{i=1}^{N} \overline{R}_i^T S_i \overline{R}_i.$$

Remark 3. *We note that $S_i \neq \overline{R}_i S \overline{R}_i^T$, since the later involves contribution from neighbor subdomains. However, given $u \in V_\Gamma^{(i)}$ then $u^T S_i u \leq u^T(\overline{R}_i S \overline{R}_i^T)u$, since S and S_i are positive definite.*

2.3.1 BDD Preconditioning

In order to present the variational setting of the preconditioner, we decompose the space V as

$$V = \oplus_{i=1}^{N} V_i \oplus V_\Gamma$$

where $V_i = H^1_0(\Omega_i)\cap V$ and

$$V_\Gamma = \{u \in V;\ u|_{\Omega_i} = \mathcal{H}^{(i)}_{\text{adv}}(u|_{\Gamma_i}) \text{ for } i = 1,\ldots,N\}.$$

We introduce the bilinear form $s(\cdot,\cdot)$ and its symmetric part $\hat{s}(\cdot,\cdot)$ defined on V_Γ as: given u, v in V_Γ

$$s(u,v) = v^T S u \tag{2.9}$$

and

$$\hat{s}(u,v) = v^T \widehat{S} u, \tag{2.10}$$

respectively, where $\widehat{S} = \frac{1}{2}(S + S^T)$ is the symmetric part of the matrix S. We use the symbol $\widehat{\ }$ over an operator or matrix to denote its symmetric part.

In order to define the local and coarse problems as projection-like operators we decompose the space V_Γ in the non-direct sum

$$V_\Gamma = \sum_{i=1}^{N} V_\Gamma^{(i)} + V_0$$

where

$$V_\Gamma^{(i)} = \{u \in V_\Gamma;\ u|_{\Gamma\setminus\Gamma_i} = 0\}$$

and V_0 is a coarse space which is defined in Subsection 2.3.3. We remak that a function v on $V_\Gamma^{(i)}$ is uniquely defined by its restriction $v|_{\Gamma_i}$ to Γ_i. By other side, a function on Γ_i is uniquely defined by the nodal values on the nodes of the discrete Γ_i, hence, sometimes we will refer a function $v \in V_\Gamma^{(i)}$ as the same as a vector defined by the nodal values of $v|_{\Gamma_i}$ Γ_i.

Our objective now is to develop a preconditioner for the Schur complement system (2.7), so that the resulting linear system is scalable and well-conditioned.

An initial attempt was initially proposed for the Poisson equation by De Roeck and Le Tallec in [43]. Their idea was to use an additive Schwarz like preconditioner of the form

$$M^{-1} = \sum_i R_i^T S_i^{-1} R_i$$

where $R_i : \Gamma \to \Gamma_i$ is a weighted restriction operator defined as $R_i = D_i^{-1}\overline{R}_i$, S_i is the Schur complement of the local stiffness matrix $A^{(i)}$, $\overline{R}_i : \Gamma \to \Gamma_i$ is the discrete restriction operator to Γ_i, and D_i^{-1} is a diagonal matrix defining a partition of unity on Γ, i.e., $\sum_{i=1}^{N} \overline{R}_i^T D_i^{-1}\overline{R}_i = I$, where I is the identity on Γ. The partition of unity may be defined through the *counting functions*, which for each subdomain Ω_i is defined as the operator $\delta_i : \Gamma_i \to \mathbb{R}$ such that $\delta_i(x) =$ number of subdomains sharing the node $x \in \Gamma_i$. Then, define $D_i = \text{diag}\,\{\delta_i\}$.

Note that this preconditioner does not have a mechanism to broadcast the local solutions among the subdomains, therefore the convergence deteriorates when the number of subdomains increases. So, a usual way to overcome this is to introduce a coarse solver in the method. This coarse solver will help us also avoid some issues concerning the solvability of the local problems, which we make it clear soon.

This modification was first proposed by Mandel in [39]. In order to present the matricial form of the preconditioner, let $V_0 \subset V_\Gamma$ be a coarse space to be defined later and let R_0^T be the extension matrix whose columns are basis vectors of the coarse space.

So, the preconditioner is defined as

$$M^{-1} = Q_0 + (I - Q_0 S) \sum_i Q_i (I - S Q_0)$$

where $Q_i = R_i^T S_i^\dagger R_i$, $Q_0 = R_0^T S_0^{-1} R_0$ and $S_0 = R_0 S R_0^T$.

Then the preconditioned operator is

$$M^{-1} S = P_0 + (I - P_0) \sum_i P_i (I - P_0)$$

where $P_0 = Q_0 S$ is a projection on the coarse space V_0 and $P_i = Q_i S$ is a projection-like operator on the local space $V_\Gamma^{(i)}$.

2.3.2 Local Problems

We introduce the local problems in a variational context as a projection-like operator $\widetilde{P}_i : V_\Gamma \to V_\Gamma^{(i)}$ as: for a given $u \in V_\Gamma$ we define its projection $\widetilde{P}_i u$ as the solution of the variational problem

$$s_i(\widetilde{P}_i u, v_i) = s(u, R_i^T v_i) \quad \forall v_i \in V_\Gamma^{(i)}.$$

From this formulation, we can obtain $\widetilde{P}_i = S_i^{-1} R_i S$. Finally, we define $P_i : V_\Gamma \to V_\Gamma$ as $P_i = R_i^T \widetilde{P}_i = R_i^T S_i^{-1} R_i S$.

We observe that to apply S_i^{-1} on a vector f_i on Γ_i is equivalent to solve the discrete problem

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{pmatrix} \begin{pmatrix} w_I \\ w_\Gamma \end{pmatrix} = \begin{pmatrix} 0 \\ f_i \end{pmatrix}, \tag{2.11}$$

where $w_\Gamma = S_i^{-1} f_i$. We point out that (2.11) is the discretization of the variational problem

$$\begin{cases} a_i(u, v) = \int_{\Gamma_i} f_i v \, \mathrm{ds} & \forall v \in H^1_{0, \partial\Omega \cap \partial\Omega_i}(\Omega_i) \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{2.12}$$

where the right hand side is the discretization of the Robin boundary condition $f_i = \partial_n u + \frac{1}{2}\boldsymbol{a}.nu$.

A *floating subdomain* Ω_i is a subdomain such that $\partial\Omega \cap \partial\Omega_i = \emptyset$. In this case, the problem (2.12) has pure Robin boundary condition. So, the linear system may have a kernel, and in this case the problem may have no solution or the solution may not be unique. The issue of existence of solution is solved with the introduction of a coarse problem in the preconditioner to enforce that f_i be in the image of the operator S_i. This is equivalent to f_i being orthogonal to the kernel of the adjoint operator S_i. So, we need to describe the kernel subspace of the adjoint operator in order to create the coarse space.

The adjoint of the bilinear form $a_i(\cdot,\cdot)$ is the bilinear form

$$\overline{a}_i(u,v) = a^{(i)}_{\text{symm}}(u,v) - a^{(i)}_{\text{skew}}(u,v).$$

Since the Lax-Friedrichs inequality holds in the subspace $\widehat{V}^{(i)} = \{v \in H^1(\Omega_i);\ \int_{\Omega_i} v\ \text{dx} = 0\}$, that is, the subspace of zero average functions in Ω_i, then the bilinear form $\overline{a}_i(\cdot,\cdot)$ is coercive on this space. Then by Lax-Milgram theorem, the discrete variational problem

$$\overline{a}_i(u,v) = 0 \quad \forall v \in \widehat{V}^{(i)}$$

has a unique solution in $\widehat{V}^{(i)}$. So, the kernel of the adjoint operator is the same as the original operator $a_i(\cdot,\cdot)$, and it is the trivial one or is the subspace of constant functions.

Remark 4. *In the case of a non-vanishing constant advection field a the operator $a_i(\cdot,\cdot)$ and its adjoint kernel have only the trivial solution. Indeed, if it were not the case, then $u = 1$ would be a basis of the kernel. However, for any $v \in V$, $a^{(i)}_{diff}(1,v) = 0$, and applying Lemma 1, we obtain that*

$$a^{(i)}_{adv}(1,v) = (\boldsymbol{a}.\nabla 1, v) + \frac{1}{2}\int_{\Gamma_i} \boldsymbol{a}.\boldsymbol{n} v\ dx = \frac{1}{2}\int_{\Gamma_i} \boldsymbol{a}.\boldsymbol{n} v\ dx,$$

which does not vanish for any $v \in V$. This implies that the kernel is the trivial one.

2.3.3 Coarse Problem

In balancing domain decomposition methods, one should pay great attention on the choice of the coarse space. The coarse problem is introduced in the preconditioner so that the local Robin problems are solvable, and furthermore to make the preconditioner scalable by broadcasting information through all subdomains.

To avoid the eventual unsolvability of the local Robin problems we take the coarse space as a scaling of the constant functions in each subdomain. This choice of the constant functions is based on the possible kernel of the adjoint of operator $a^{(i)}(\cdot,\cdot)$, which might include the constant functions. The weight functions used to define the local problems forms the basis of the coarse space.

Then, we define the coarse space as

$$V_0 = \text{span}\ \{R_i^T\mathbf{1};\ i = 1,\ldots,N\}$$

where $\mathbf{1} \in V_\Gamma^{(i)}$ is the vector of ones. We define the restriction operator R_0 as the matrix whose rows are the vectors that span the coarse space V_0, i.e., the vectors $R_i^T\mathbf{1}$, for $i = 1,\ldots,N$.

The projection operator $P_0 : V_\Gamma \to V_0$ is defined for a given $u \in V_\Gamma$ as the vector P_0u solution of the discrete variational problem

$$s(P_0u, v) = s(u, v) \quad \forall v \in V_0.$$

A simple computation gives the matricial form of the projection

$$P_0 = R_0^T S_0^{-1} R_0 S$$

where $S_0 = R_0 S R_0^T$. We remark that the set $\{R_i^T\mathbf{1};\ i = 1,\ldots,N\}$ can be linearlly dependent, and then S_0 becomes singular. To see this, just take the square domain $[0,1]\times[0,1]$ and decompose it into square subdomains. In order to avoid this drawback one can eliminate the coarse function associated to a non-floating subdomain.

Lemma 5. *Suppose that the columns of R_0^T are linearlly independent. Then, given $u \in V_\Gamma$,*

$$P_0u = 0$$

if and only if

$$< Su, R_i^T\mathbf{1} >= 0, \quad \forall i = 1,\ldots,N.$$

Proof. Just note that the kernel of R_0^T is the trivial kernel, S_0 is invertible, and that

$$R_0^T S_0^{-1} R_0 Su = P_0u = 0$$

is equivalent to $(R_i^T\mathbf{1})^T Su = 0, \quad \forall i = 1,\ldots,N.$ □

We define the subspace of V_Γ that is S-orthogonal to V_0 as

$$V_0^\perp = \{v \in V_\Gamma;\ s(v,w) = 0 \quad \forall w \in V_0\}.$$

In particular, by definition of P_0, for any $u \in V_\Gamma$ we have $(I-P_0)u \in V_0^\perp$. Hence, for any $v \in V_0^\perp$ we obtain that $P_0v + (I-P_0)\sum_{i=1}^N P_i(I-P_0)v = (I-P_0)\sum_{i=1}^N P_iv$.

2.4 BDDC - Balancing Domain Decomposition with Constraints

The balancing domain decomposition with constraints, known as BDDC, shares some similarities with the well-known BDD methods presented earlier, and the FETI and FETI-DP methods(see [37, 36]). FETI-DP and BDDC methods do not require the solution of linear systems with singular matrices, as is the case of the BDD and one-level FETI([18, 17]). The similarities of the BDDC method and the Neumann-Neumann methods are verified for plate and shell problems [50]. Depite the similarities, there are the differences. Just to mention, in the BDDC method the primary variables for iterative solution are displacements while in FETI-DP they are Lagrange multipliers. The BDD method uses a multiplicative coarse grid correction, while the BDDC method uses an additive one. In BDDC, constraints are introduced on the coarse and local spaces, decomposing them into a direct sum. The constraints also make the local spaces solvable even on the floating subdomain case.

For each subdomain Ω_i we introduce constraints on the variables on Γ_i through a constraint matrix C_i of size $n_{c_i} \times n_i$, where n_{c_i} is the number of local coarse degrees of freedom and n_i is the number of degrees of freedom on Γ_i.

To fix ideas, let us consider the vertex based BDDC. On each substructure Ω_i we mean by a corner the vertex of Γ_i that shares more than two subdomains, and let us denote by $\mathcal{N}_i$ the collection of such nodes on Γ_i. In the vertex based BDDC, n_{c_i} is the dimension of $\mathcal{N}_i$ and for each vector $u_\Gamma^{(i)}$ on Γ_i, the vector $C_i u_\Gamma^{(i)}$ will be the values of $u_\Gamma^{(i)}$ on the nodes in $\mathcal{N}_i$.

In the BDDC methods, two consistency conditions are required:

1. a local space consistency with the weight matrices in the sense that: *for any* $u_i \in \Gamma_i$ *such that* $C_i u_i = 0$ *then*

$$C_j \overline{R}_j \overline{R}_i^T D_i^{-1} u_i = 0 \quad \forall j = 1, \cdots, N.$$

2. a coarse space consistency between substructures in order to define an interpolation in the sense that *for any* v *on* Γ*, there exists* u_c *on* $\mathbb{R}^{n_c}$ *such that*

$$C_i \overline{R}_i v = R_{c_i} u_c \quad \forall i = 1, \cdots, N.$$

As we will see in the next subsections, the vertex constraint will impose Dirichlet condition at the corner vertices on the local problems and coarse basis functions. We consider the coarse and local problems in a general framework, however we particularly present C_i given by the vertex based BDDC.

2.4.1 The Coarse Problem

On each substructure Ω_i we define the local coarse basis functions $\Phi_i^j = w_\Gamma$ ($j = 1, \cdots, n_{c_i}$, where n_{c_i} is the number of local coarse degrees of freedom), as the solutions of

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} & 0 \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} & C_i^T \\ 0 & C_i & 0 \end{pmatrix} \begin{pmatrix} w_I \\ w_\Gamma \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ e_j \end{pmatrix} \tag{2.13}$$

where e_j is the column j of the identity matrix of size n_{c_i}. Let us denote by $\Phi_i = [\Phi_i^1 \cdots \Phi_i^{n_{c_i}}]$ the matrix whose columns are the local coarse basis functions. Hence, in the case of the vertex based BDDC, this is a problem with a Dirichlet condition set to one on the node j and zero in the other nodes of $\mathcal{N}_i$, and zero Neumann data on the remaining interface nodes.

The local coarse spaces $V_0^{(i)}$ are defined as

$$V_0^{(i)} = \{u_0 \in \Gamma_i;\ u_0 = \Phi_i u_{c_i} \text{ and } u_{c_i} \in \mathbb{R}^{n_{c_i}}\}.$$

In the case of the vertex based BDDC, the second consistency condition on the constraints imposes continuity on the corner vertex between neighboring coarse spaces $V_0^{(i)}$. In fact, this consistency condition imposes continuity on the primal variables.

In the BDDC preconditioning method, the assembling of the coarse matrix S_0^c is calculated as

$$S_0^c = \sum_{i=1}^{N} R_{c_i}^T \Phi_i^T S_i \Phi_i R_{c_i},$$

where it is interesting to observe that its assembling can be performed on the subdomains with very few communication steps between the subdomains, since its computation involves only the substructures that share the coarse degree(s) of freedom.

The coarse problem is then defined as

$$Q_0 = R_0^T S_0^c R_0 \tag{2.14}$$

where R_0 is the restriction matrix of the degrees of freedom on Γ to the coarse degrees of freedom defined as

$$R_0 = \sum_{i=1}^{N} R_{c_i}^T \Phi_i^T D_i^{-1} \overline{R}_i$$

with R_{c_i} the zero-one matrix which maps the global coarse to local coarse degrees of freedom.

We also note that S_0^c is different of the coarse matrix $S_0 = R_0 S R_0^T$ whose computation is more expensive, since its computation include the neighbor of the neighbor subdomains of the subdomains involved in the computation of S_0^c.

2.4.2 The Local Problems

The local problems are defined on the dual of the coarse space variables. The local spaces are defined as

$$V_\Gamma^{(i)} = \{u_i \in \Gamma_i;\ \ C_i u_i = 0\}.$$

The first consistency condition of the BDDC will impose a consistency on the weightings D_i^{-1}, in the sense that if u_i is in $V_\Gamma^{(i)}$ then for any $j = 1, \cdots, N$ the vector $\overline{R}_j \overline{R}_i^T D_i^{-1} u_i$ is in $V_\Gamma^{(j)}$.

We define local problems as

$$Q_i = \overline{R}_i^T D_i^{-1} S_i^{c^{-1}} D_i^{-1} \overline{R}_i \tag{2.15}$$

where $u_\Gamma^{(i)} = S_i^{c^{-1}} \tilde{b}_i$ is defined as the solution of

$$\begin{pmatrix} S_i & C_i^T \\ C_i & 0 \end{pmatrix} \begin{pmatrix} u_\Gamma^{(i)} \\ \lambda_i \end{pmatrix} = \begin{pmatrix} \tilde{b}_i \\ 0 \end{pmatrix}$$

which is equivalent to solving

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma}^{(i)} & 0 \\ A_{\Gamma I}^{(i)} & A_{\Gamma\Gamma}^{(i)} & C_i^T \\ 0 & C_i & 0 \end{pmatrix} \begin{pmatrix} u_I^{(i)} \\ u_\Gamma^{(i)} \\ \lambda_i \end{pmatrix} = \begin{pmatrix} 0 \\ \tilde{b}_i \\ 0 \end{pmatrix}. \tag{2.16}$$

The reader can note that the only difference of the local problems here to the ones of the BDD method is the modification of the local Schur complement by introducing the constraints as Lagrange multipliers. In the vertex based BDDC, the constraints impose zero Dirichlet condition on the nodes in $\mathcal{N}_i$.

2.4.3 The Constraints and Implementation Issues

In order to present some usual constraints, let us introduce some new notation. A corner vertex is a vertex that shares more than two subdomains. An edge between two substructures Ω_i and Ω_j is the collection of vertices on a connected segment of $\overline{\Omega}_i \cap \overline{\Omega}_j$. The collection of corners of a substructure Ω_i will be denoted as $\mathcal{N}_i$ and the collection of the edges as $\mathcal{E}_i$.

We consider the following constraints:

1. fix the corner values; or
2. fix the averages on the edges.

As we have presented before, the corner values constraints impose a Dirichlet condition in the local problems and in the definition of the coarse basis functions through the constraint matrix C_i, where $C_i u_i$ is the value of u_i at the corner nodes of the substructure Ω_i. These constraints make the local problems (2.16) solvable even in the floating subdomain case, hence, we do not need an extra coarse problem to make them solvable, as it was in the case in the balancing Neumann-Neumann preconditioner.

The introduction of the vertex constraint is made by imposing the Dirichlet condition in the linear system. We define $\Gamma_\Pi = \mathcal{N}_i$ and $\Gamma_\triangle = \Gamma \backslash \Gamma_\Pi$ where Π stands for *primal* and $\triangle$ stands for *dual*. Thus the local problems (2.16) turn to be:

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma_\triangle}^{(i)} \\ A_{\Gamma_\triangle I}^{(i)} & A_{\Gamma_\triangle\Gamma_\triangle}^{(i)} \end{pmatrix} \begin{pmatrix} u_I^{(i)} \\ u_{\Gamma_\triangle}^{(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ \tilde{b}_i \end{pmatrix}$$

and the coarse functions $\Phi_i^j = \begin{pmatrix} u_{\Gamma_\triangle}^{(i)} \\ e_j \end{pmatrix}$ are computed as

$$\begin{pmatrix} A_{II}^{(i)} & A_{I\Gamma_\triangle}^{(i)} \\ A_{\Gamma_\triangle I}^{(i)} & A_{\Gamma_\triangle\Gamma_\triangle}^{(i)} \end{pmatrix} \begin{pmatrix} u_I^{(i)} \\ u_{\Gamma_\triangle}^{(i)} \end{pmatrix} = \begin{pmatrix} -A_{I\Gamma_\Pi}^{(i)} e_j \\ -A_{\Gamma_\triangle\Gamma_\Pi}^{(i)} e_j \end{pmatrix},$$

where e_j is the column j of the identity matrix of size $n_{c_i} \times n_{c_i}$.

In the edge based version, we impose an average on each edge. To each row j of C_i we associate an edge E_j. The matrix C_i is then defined such that the element of line j of the vector $C_i u_i$ is the average of the vector u_i on the edge. Then, in a structured mesh, the rows of C_i are zero-one entries, with one in the columns associated to the nodes of the associated edge.

2.4.4 Matricial Form of the Preconditioner

The local coarse spaces $V_0^{(i)}$ and the local spaces V_i provide a direct sum decomposition of the space V_Γ in the following way

$$V_\Gamma = \widetilde{R}_0^T V_0 \oplus \sum_{i=1}^{N} R_i^T V_i$$

where $V_0 = \prod_{i=1}^{N} V_0^{(i)}$ is discontinuous on the dual degrees of freedom and continuous on the primal degrees of freedom. For each $v = (v_i)_{i=1}^{N} \in V_0$ we define the extension operator $\widetilde{R}_0^T v = \sum_{i=1}^{N} \overline{R}_i^T D_i^{-1} v_i$.

The preconditioner has a two-level additive Schwarz preconditioner form, that is,

$$M^{-1} = \sum_{i=1}^{N} Q_i + Q_0$$

where Q_i is given in (2.15) and Q_0 is given in (2.14). Thus, the prencoditioned operator is

$$M^{-1}S = \sum_{i=1}^{N} Q_i S + Q_0 S$$

2.5 Numerical Results

For the numerical experiments we consider the domain as the square $[0,1] \times [0,1]$ decomposed into $\sqrt{N} \times \sqrt{N}$ quadrilateral subdomains, as sketched in Figure 2.1. Since the linear problems we are dealing with are non-symmetric we consider the preconditioned GMRES method as the iterative solver. The method iterates until we obtain the relative residual $\|M^{-1}r_n\|_2 / \|M^{-1}r_0\|_2$ of the preconditioned system less than 10^{-6}.

The model problem we consider is

$$-\nu \triangle u + \boldsymbol{a}.\nabla u = 0 \quad \text{on } \Omega$$

with the Dirichlet boundary condition

$$u(x,y) = \begin{cases} 0 & \text{for } (x,y) \in \zeta_1 \\ 1 & \text{for } (x,y) \in \zeta_2 \end{cases}$$

where ζ_1 and ζ_2 is the decomposition of $\partial\Omega$ as sketched in Figure 2.2.

Figure 2.1: Sketch of mesh and subdomains

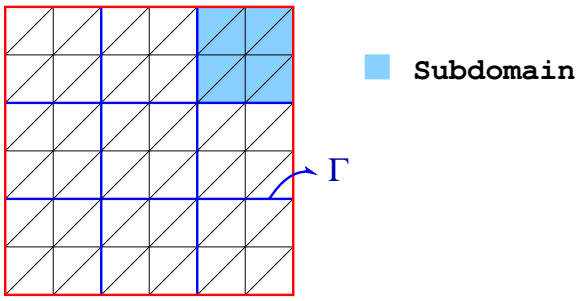
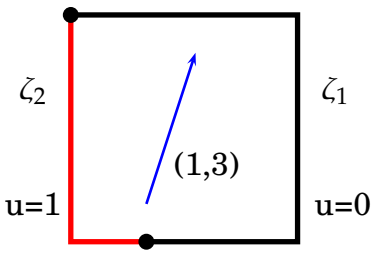


Figure 2.2: Dirichlet condition of test problem.



On Tables 2.1, 2.2, 2.3 and 2.4 we fix the global mesh to $h = 1/512$ and the element Péclet number to $\mathrm{Pe}_K = 0.145$, $\mathrm{Pe}_K = 1.45$, $\mathrm{Pe}_K = 14.5$ and $\mathrm{Pe}_K = 145$, respectively. The definition of Pe_K is given in (2.6). On these tables we compare the preconditioners

1. the BDDC with corner vertex constraints.
2. the BDDC with edge average constraints.
3. the BDD with Q_1 functions on the coarse mesh as the coarse space.
4. the BDD with constant by subdomain coarse space as presented in Section (2.3).
5. the multiplicative BDDC where we consider as coarse problem

$$Q_0 = R_0^T S_0 R_0$$

with $S_0 = R_0 S R_0^T$, and the preconditioner as

$$M^{-1} = Q_0 + (I - Q_0 S) \sum_{i=1}^{N} Q_i (I - S Q_0)$$

We compare the preconditioners for different number of subdomains and local meshes which we present in parenthesis.

The balancing Neumann-Neumann Preconditioner (4) presents stability in the iterations in Tables 2.1 and 2.2 when we increase the number of subdomains, however the Tables 2.3 and 2.4 show that the iterations increase as we increase the Péclet number of the problem, more notably in the case of 64×64 subdomains. Hence, one can conclude that the balancing Neumann-Neumann preconditioner performs well in the diffusive dominated models only.

On Tables 2.1, 2.2, 2.3 and 2.4 we can note that the edge based BDDC (preconditioner (2)) presents a better behavior than preconditioner (4). However, these two preconditioners present the same sensibility on the Péclet number.

The Preconditioners (1), (3) and (5) present a small sensibility on the Péclet number as can be noticed on Tables 2.1, 2.2, 2.3 and 2.4.

The BDD with a Q_1 coarse space (Preconditioner (3)) presents a good behavior on the Pèclet number and on the number of subdomains. However it is not suitable for unstructured meshes.

The Preconditioner (5) presents a very good behavior, presenting in almost all the tests the better results. However, its efficient implementation is hard and the preconditioner presents much more communication than preconditioner (1).

For the numerical tests we consider the Preconditioner (1). These results are presented on Tables 2.5 and 2.6. We remark that the preconditioner is not sensible to the direction of the advection field, thus the results of our numerical experiments for the constant advection $\boldsymbol{a} = (1, 3)$ provides the behavior of the preconditioner for any direction of the constant advection field.

On Table 2.5 we fix the local mesh on the substructures and increases the number of subdomains in order to analyze the scalability of the preconditioner. As we can see the preconditioner is not sensible to the size of the local meshes. On the other hand, the number of iterations depends on the number of subdomains in a coordinate direction, i.e., on $1/H$. This behavior can be better understood on the Figures 2.5 and 2.6 where we plot the first 4 intermediate solutions of the iterative solver. These plots are in the case of a local mesh 8×8 and with 4×4 subdomains as in line 3 of Table 2.5. On these figures we observe that the boundary condition "travels" in the direction of the advection, subdomain by subdomain, which explains the number of iterations. We observe that this behavior does not happen in the preconditioners that have a coarse problem treated multiplicatively, that is, in preconditioners (3), (4) and (5).

In Figures 2.3 and 2.4 we plot the decay of the relative residuals in the cases that we fix the local mesh to 16×16 and when we fix the number of subdomains to 8×8, respectively. One can note that the residuals do not decrease much until the number of iterations is order of number of subdomains in the coordinate direction, i.e., when the boundary solution has travelled on the domain. After that, the residual decays very fast.

In Table 2.6 we fix the number of subdomains and refine the local mesh of the substructures. In this case we can see the stability of the preconditioner with respect to the local mesh for a fixed number of subdomains.

2.6 Conclusions

In this chapter we extend the use of the BDD and BDDC preconditioners to stabilized advection-diffusion equations, trying different coarse

Table 2.1: Preconditioned GMRES iteration counts for viscosity equal to 0.01 and the advection field $(1,3)$, which gives $\mathrm{Pe} = 316$, defined in (2.5), and $\mathrm{Pe}_K = 0.145.$, defined in (2.6). We fix the global mesh to 512×512, and change the number of subdomains and the local mesh (presented in parenthesis).

		Subdomains (local mesh)			
		8 (64)	16 (32)	32 (16)	64 (8)
Preconditioner	(1)	12	20	33	52
	(2)	17	20	15	13
	(3)	13	19	33	58
	(4)	16	17	15	12
	(5)	9	14	21	21

Table 2.2: Preconditioned GMRES iteration counts for viscosity equal to 0.001 and the advection field is (1,3), which gives $\mathrm{Pe} = 3162$ and $\mathrm{Pe}_K = 1.45$. We fix the global mesh to 512×512, and change the number of subdomains and the local mesh (presented in parenthesis).

		Subdomains (local mesh)			
		8 (64)	16 (32)	32 (16)	64 (8)
Preconditioner	(1)	9	17	30	53
	(2)	13	21	32	37
	(3)	12	19	34	46
	(4)	18	26	32	30
	(5)	7	13	24	34

Table 2.3: Preconditioned GMRES iteration counts for the viscosity is 0.0001 and the advection field is (1,3), which gives Pe $= 31620$ and $\mathrm{Pe}_K = 14.5$. We fix the global mesh to 512×512, and change the number of subdomains and the size of local mesh (in parenthesis).

		Subdomains (local mesh)			
		8 (64)	16 (32)	32 (16)	64 (8)
Preconditioner	(1)	9	15	28	53
	(2)	15	23	38	63
	(3)	12	19	34	47
	(4)	19	33	46	67
	(5)	8	13	24	46

Table 2.4: Preconditionerd GMRES iteration counts for the viscosity is 0.00001 and the advection field is (1,3), which gives Pe $= 316200$ and $\mathrm{Pe}_K = 145$. We fix the global mesh to 512×512, and change the number of subdomains and the size of local mesh (presented in parenthesis).

		Subdomains (local mesh)			
		8 (64)	16 (32)	32 (16)	64 (8)
Preconditioner	(1)	9	16	29	54
	(2)	16	24	41	70
	(3)	13	19	32	49
	(4)	20	35	58	75
	(5)	8	13	23	45

Table 2.5: Preconditioned GMRES iteration counts and within parenthesis is the local Péclet number. We fix the local mesh size to H/h equal to 8, 16 and 32. We consider $a = (1,3)$ and $\nu = 0.00001$ (Pe $= 3.16e+5$).

		H/h		
		8	16	32
Subdomains	2×2	4 (4.6e+3)	3 (2.3e+3)	3 (1.2e+3)
	3×3	6 (3.1e+3)	5 (1.6e+3)	4 (7.8e+2)
	4×4	7 (2.3e+3)	7 (1.2e+3)	6 (5.8e+2)
	8×8	11 (1.2e+3)	10 (5.8e+2)	10 (2.9e+2)
	12×12	15 (7.8e+2)	14 (3.8e+2)	13 (1.9e+2)
	16×16	18 (5.8e+2)	17 (2.9e+2)	16 (1.4e+2)
	32×32	31 (2.9e+2)	29 (1.4e+2)	28 (7.3e+1)

Table 2.6: Preconditioned GMRES iteration counts and within parenthesis is the local Péclet number. Fixing the number of subdomains to 8×8 and 12×12. We consider $a = (1,3)$ and $\nu = 0.00001$ (Pe $= 3.16e+5$).

		Subdomains	
		8×8	12×12
Local mesh	4×4	14 (2.3e+3)	17 (1.5e+3)
	8×8	11 (1.2e+3)	15 (7.8e+2)
	16×16	10 (5.8e+2)	14 (3.9e+2)
	32×32	10 (2.9e+2)	13 (1.9e+2)
	64×64	9 (1.4e+2)	12 (9.7e+1)
	128×128	9 (7.2e+1)	12 (4.8e+1)

Figure 2.3: Residuals for local mesh 16×16. We consider $\boldsymbol{a} = (1,3)$ and $\nu = 0.00001$, thus $\mathrm{Pe} = 3.16e+5$.

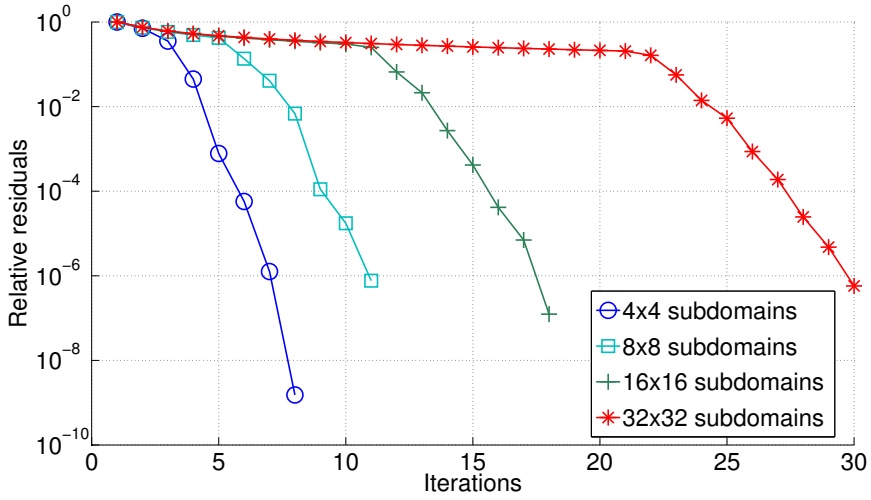


Figure 2.4: Residuals for 8×8 subdomains. We consider $\boldsymbol{a} = (1,3)$ and $\nu = 0.00001$, thus $\mathrm{Pe} = 3.16e+5$.

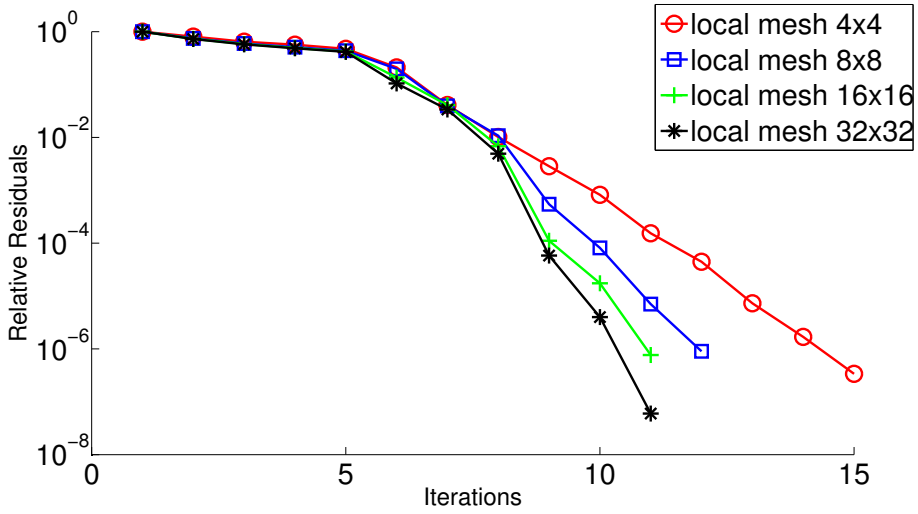


Figure 2.5: Iterative solutions for 4×4 subdomains and local mesh 8×8. Considering $\nu = 0.00001$ and advection field (1,3).

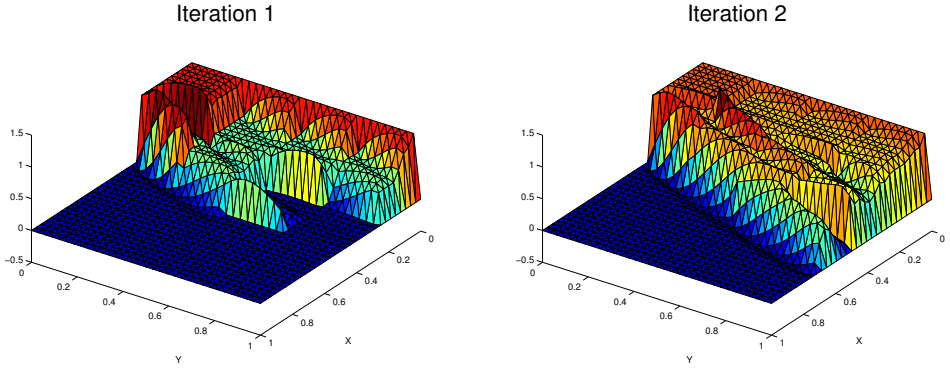
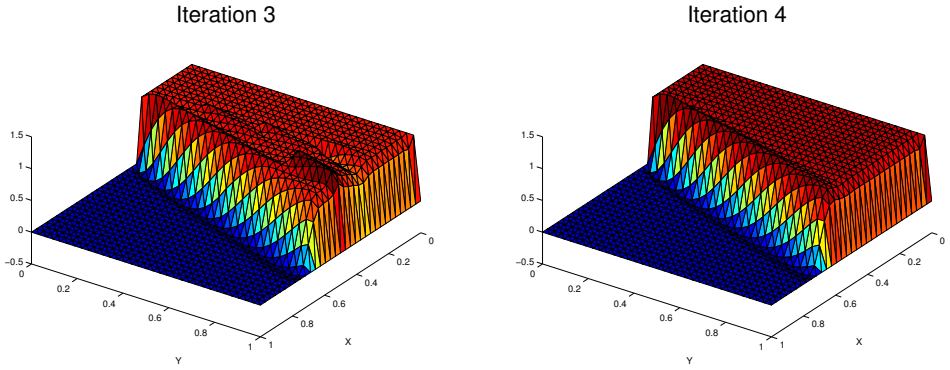


Figure 2.6: Iterative solutions for 4×4 subdomains and local mesh 8×8. Considering $\nu = 0.00001$ and advection field (1,3).



spaces. In all the numerical experiments performed in the earlier section we also tryied some different weighting matrices without increating performace. In the tests we verify the behavior of the different solvers with respect to the local Péclet number (2.6), the number of subdomains, and the local mesh. The BDD with a coarse spaces of constant functions was proven to be a good alternative for local Péclet number of order 1. However the BDD with a coarse space of Q1 functions was proven to be a good alternative for high local Péclet number, it is unsuitable for unstructured meshes. The BDDC using edges constraints has been proven to have similar behavior as the BDD with coarse space of constant functions, thus, it is also a good alternative for local Péclet numbers of order 1.

The BDDC by vertex constraints has been shown to have a good performance for high local Péclet numbers, since the iterations seems to depend on the number of subdomains in the direction of the advection field, and to independ on the local Péclet number. The numerical experiments with this preconditioner shows that the broadcast of the solution is performed in the direction of the streamlines, which indicates why the number of iterations depends on the number of subdomains in the streamline direction.

We also considered the BDDC by vertex constraint with a multiplicative aproach, which shows a better performance than the additive approach. However, this preconditioner has the drawback that its implementation is more costly.

CHAPTER 3

BNN for Stokes equations

3.1 Introduction

The goal of this chapter is to introduce several improvements of the Pavarino and Widlund method which are essential for its efficient application. We are particularly concerned with aspects associated to unstructured mesh parallel implementation and the high cost of the subdomain solvers when high-order Stokes discretizations are considered. We introduce several possible choices for unstructured coarse spaces and discuss their advantages in terms of scalability, implementation efforts and robustness with respect to the coefficient jumps. With regards to the high cost of the subdomain solvers, we explore how the inf-sup condition of Stokes discretization are checked in order to perform proper element-wise static condensation and decrease the number of interior unknowns. We show that the computational complexity of the two discretizations, the higher-order $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ and the lower-order $\mathbf{P_2}/\mathrm{P}_0$ elements, have comparable computational costs. The chapter is organized as follows. Sections 3.2 and 3.3 present the Stokes equations and the variational formulation, respectively, while on Section 3.4 we introduce the discretizations used in the numerical experiments. Section 3.5 is devoted to the BDD preconditioner for the Stokes equations and the coarse spaces. In Section 3.6 we discuss some of the implementation issues, and in Section 3.7 we provide numerical results.

3.2 The Stokes Model

Let $\Omega \subset \mathbb{R}^2$ be a domain with a polygonal boundary. Given $A = (a_{ij})$ and $B = (b_{ij})$ in $M(n)$ we define the product

$$(A : B) = \sum_{i,j=1}^{n} a_{ij} b_{ij}$$

where $M(n)$ is the set of $n \times n$ matrices with entries in $\mathbb{R}$.

For a function $w \in L^2(\Omega)$ denote $\|w\|_0^2 = \int_\Omega w^2 \, \mathrm{dx}$, and for a function $\boldsymbol{v} \in H^1(\Omega)^2$ denote

$$|\boldsymbol{v}|_1^2 = \int_\Omega (\nabla \boldsymbol{v} : \nabla \boldsymbol{v}) \, \mathrm{dx} \quad \text{and} \quad |\boldsymbol{v}|_{\mathrm{div}}^2 = \int_\Omega (\nabla \cdot \boldsymbol{v})^2 \, \mathrm{dx} = \|\nabla \cdot \boldsymbol{v}\|_0^2.$$

Consider the Stokes equations:

$$\left\{ \begin{array}{rcll} -2\nu \nabla \cdot \boldsymbol{\varepsilon}(\boldsymbol{u}) + \nabla p &=& \boldsymbol{f} & \text{in } \Omega \\ -\nabla \cdot \boldsymbol{u} &=& g & \text{in } \Omega \\ \boldsymbol{u} &=& \boldsymbol{u}_d & \text{on } \partial\Omega \end{array} \right. \tag{3.1}$$

where $\nu > 0$ is the kinematic viscosity and the $\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}[\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T]$ denotes the symmetric stress tensor. In this chapter, we assume only Dirichlet boundary conditions with the compatibility condition

$$\int_\Omega -g \, \mathrm{dx} = \int_{\partial\Omega} \boldsymbol{u}_d \cdot \boldsymbol{n} \, \mathrm{ds}.$$

The treatment of natural boundary conditions is similar and does not bring any extra difficulties; see also Remark 8.

Remark 6. *Since we are assuming Dirichlet boundary conditions on the whole boundary $\partial\Omega$, the velocity solution is unique and the pressure is unique up to a constant. To make the pressure unique, we impose the additional condition of zero average pressure on Ω, i.e., $\int_\Omega p \, dx = 0$.*

3.3 Variational Formulation

The variational formulation is introduced as follows. Let us define the space of velocities $\mathbf{X} = H_0^1(\Omega)^2$ and the space of pressures $M = L_0^2(\Omega)$, where $L_0^2(\Omega)$ stands for $L^2(\Omega)$ functions with zero average in Ω. Given

$\boldsymbol{f} \in H^{-1}(\Omega)^2$ and $g \in L^2(\Omega)$, the variational formulation of the Stokes equations is given by:

Find $\boldsymbol{u} \in \mathbf{X}$ *and* $p \in M$ *such that*

$$\begin{cases} a(\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) & = F(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathbf{X}, \\ b(\boldsymbol{u},q) & = G(q) \quad \forall q \in M, \end{cases} \tag{3.2}$$

where $a(\boldsymbol{u},\boldsymbol{v}) = 2\nu(\varepsilon(\boldsymbol{u}) : \varepsilon(\boldsymbol{v}))_\Omega$, $b(\boldsymbol{v},p) = -(\nabla \cdot \boldsymbol{v}, p)_\Omega$, $F(\boldsymbol{v}) = (\boldsymbol{f},\boldsymbol{v})_\Omega$, and $G(q) = (g,q)_\Omega$. The solution $(\boldsymbol{u},p) \in \mathbf{X} \times M$ of (3.2) exists and is unique; see [22].

3.4 Discretization

Let $\mathcal{T}_h$ be a regular triangulation of Ω. We consider the mixed finite elements $\mathbf{P_2}/\mathrm{P}_0$ and $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$, where the velocity is taken continuous and the pressure discontinuous.

The $\mathbf{P_2}/\mathrm{P}_0$ mixed finite elements is described as follows: the velocity space be given as

$$\mathbf{X}_h = \{\boldsymbol{v} \in \mathbf{X};\ \boldsymbol{v}|_K \in P_2(K)^2,\ \forall K \in \mathcal{T}_h\}$$

while the pressure space by discontinuous piecewise constant functions

$$M_h = \{q \in M;\ q|_K \in P_0(K),\ \forall K \in \mathcal{T}_h\}.$$

To obtain better accurate results we consider the $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ mixed finite element space. This space can be considered as a stabilization of the unstable space $\mathbf{P_2}/\mathrm{P}_1$. We take the bubble function as $\hat{b}(\hat{x},\hat{y}) = \hat{x}\hat{y}(1-\hat{x}-\hat{y})$ defined on the reference element $\widehat{K}$, and then for each element K in $\mathcal{T}_h$ define $b_K(x,y) = \hat{b}(F_K^{-1}(x,y))$, where F_K is the affine mapping from $\widehat{K}$ to K. The velocity space $\mathbf{X}_h$ is then given as

$$\mathbf{X}_h = \{\boldsymbol{v} \in \mathbf{X};\ \boldsymbol{v} = \boldsymbol{v}_P + \boldsymbol{v}_B,\ \text{s.t. } \boldsymbol{v}_{P|_K} \in P_2(K)^2,\ \boldsymbol{v}_{B|_K} \in \mathbf{X}_B(K),\ \forall K \in \mathcal{T}_h\},$$

where for each element $K \in \mathcal{T}_h$

$$\mathbf{X}_B(K) = \{\boldsymbol{v}_B \in H_0^1(K)^2;\ \boldsymbol{v}_B = \begin{pmatrix} \alpha_1 b_K \\ \alpha_2 b_K \end{pmatrix} \text{ and } \alpha_1,\ \alpha_2 \in \mathbb{R}\}.$$

The discrete pressure space consists of discontinuous piecewise linear functions denoted by P_1 given as:

$$M_h = \{p \in M;\ p_{|_K} \in P_1(K),\ \forall K \in \mathcal{T}_h\}.$$

The two discretizations above satisfy the uniform *inf-sup* condition [22], i.e., there exists a constant β (independent of h) such that

$$\sup_{\substack{\boldsymbol{v}\in\mathbf{X}_h \\ v\neq 0}} \frac{(\nabla\cdot\boldsymbol{v},q)}{\|\boldsymbol{v}\|_{\mathbf{H^1}}} \geq \beta\|q\|_0 \quad \forall q\in M_h. \tag{3.3}$$

The inf-sup stability of the discretization $(\mathbf{P_2}+\mathbf{Bubbles})/\mathrm{P}_1$ can be shown by using the macroelement technique [48, 49]. To verify this, just consider as macroelement the union of the two elements $M = \{K_{\mathrm{ref}}, \widetilde{K}\}$ where K_{ref} is the element of reference and $\widetilde{K}$ is the element whose vertices are $(1,0)$, $(1,1)$ and $(0,1)$.

Thus, the discrete variational formulation of the Stokes problem (3.1) given by:

Find $\boldsymbol{u}\in\mathbf{X}_h$ *and* $p\in M_h$ *such that*

$$\begin{cases} a(\boldsymbol{u},\boldsymbol{v})+b(\boldsymbol{v},p) &= F(\boldsymbol{v}) \quad \forall \boldsymbol{v}\in\boldsymbol{X}_h, \\ b(\boldsymbol{u},q) &= G(q) \quad \forall q\in M_h, \end{cases} \tag{3.4}$$

has a unique solution (see [22]). In matricial form, the discrete linear system (3.4) is of the form

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}\begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}. \tag{3.5}$$

3.5 BDD for Stokes Problem

In this section we present the matricial form of the preconditioner. Decompose the domain Ω into N non-overlapping connected subdomains Ω_i and let $\Gamma = (\cup_{i=1}^N \partial\Omega_i)\backslash\partial\Omega$, then we have

$$\Omega = \cup_{i=1}^N \Omega_i \cup \Gamma\ .$$

We denote the nodes inside Ω_i by Ω_i^h, the nodes on Γ by Γ_h and the nodes on $\partial\Omega_i\cap\Gamma$ by $\Gamma_h^{(i)}$.

3.5.1 Schur Complement System

In order to perform a static condensation of the interior variables on Ω_i we reorder and denote the variables as follows:

- $\boldsymbol{u}_I$ $\leftarrow$ interior velocities
- p_I $\leftarrow$ pressures with zero average in each subdomain Ω_i
- $\boldsymbol{u}_\Gamma$ $\leftarrow$ interface velocities
- p_0 $\leftarrow$ constant pressure in each Ω_i and with zero average in Ω

Using this reordering, the matrix of the discrete system (3.5) can be written as:

$$K = \begin{pmatrix} K_{II} & K_{I\Gamma} \\ K_{\Gamma I} & K_{\Gamma\Gamma} \end{pmatrix} = \begin{pmatrix} A_{II} & B_{II}^T & A_{I\Gamma} & B_{0I}^T \\ B_{II} & 0 & B_{I\Gamma} & 0 \\ A_{I\Gamma} & B_{I\Gamma}^T & A_{\Gamma\Gamma} & B_0^T \\ B_{0I} & 0 & B_0 & 0 \end{pmatrix}.$$

The submatrix B_{0I} is null since by the divergence theorem, $\int_{\Omega_i} \nabla \cdot \boldsymbol{u}_I \, \mathrm{dx} = 0$. Eliminating the interior variables $\boldsymbol{u}_I$ and p_I by static condensation we obtain the following Schur complement system:

$$S \begin{pmatrix} \boldsymbol{u}_\Gamma \\ p_0 \end{pmatrix} = \begin{pmatrix} \widetilde{\boldsymbol{f}}_\Gamma \\ \widetilde{g}_0 \end{pmatrix}, \tag{3.6}$$

where

$$S = K_{\Gamma\Gamma} - K_{\Gamma I} K_{II}^{-1} K_{I\Gamma} = \begin{pmatrix} S_\Gamma & B_0^T \\ B_0 & 0 \end{pmatrix},$$

and

$$\begin{pmatrix} \widetilde{\boldsymbol{f}}_\Gamma \\ \widetilde{g}_0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_\Gamma \\ g_0 \end{pmatrix} - K_{\Gamma I} K_{II}^{-1} \begin{pmatrix} \boldsymbol{f}_I \\ g_I \end{pmatrix}.$$

Remark 7. *Since A_{II} is positive definite (by Korn's inequality) and B_{II} has full row rank, the K_{II} is invertible. We note also that since B_{0I} is null, it is not possible to eliminate p_0.*

Having solved the linear system (3.6), we can obtain the solutions $\boldsymbol{u}_I$ and p_I by solving

$$\begin{pmatrix} \boldsymbol{u}_I \\ p_I \end{pmatrix} = \begin{pmatrix} A_{II} & B_{II}^T \\ B_{II} & 0 \end{pmatrix}^{-1} \left[\begin{pmatrix} \boldsymbol{f}_I \\ g_I \end{pmatrix} - \begin{pmatrix} A_{I\Gamma} & 0 \\ B_{I\Gamma} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_\Gamma \\ p_0 \end{pmatrix} \right],$$

where we observe that $\boldsymbol{u}_I$ and p_I do not depend on p_0. After a reordering of the interior variables by subdomain we obtain

$$K_{II} = \begin{pmatrix} K_{II}^{(1)} & & 0 \\ & \ddots & \\ 0 & & K_{II}^{(N)} \end{pmatrix}.$$

This shows that the subdomain matrices $K_{II}^{(i)}$ are decoupled and then to apply K_{II}^{-1} to a vector is equivalent to solve N decoupled saddle problems in parallel. Notice that the multiplication by ${K_{II}^{(i)}}^{-1}$ represents a discrete

Stokes problem with Dirichlet velocity data on $\Gamma_h^{(i)}$. This solution exists and is unique since we consider the space of pressure and test functions q_I with zero average on Ω_i. Given u_{Γ_i} defined on Γ_i we define the local Stokes harmonic extension as

$$\mathcal{SH}^{(i)}(u_{\Gamma_i}) = \begin{cases} -K_{II}^{(i)^{-1}} K_{I\Gamma}^{(i)}(u_{\Gamma_i}) \\ u_{\Gamma_i} \end{cases}$$

The velocity component of $\mathcal{SH}^{(i)}$ is denoted by $\mathcal{SH}_u^{(i)}$ and the pressure component by $\mathcal{SH}_p^{(i)}$.

We also define

$$\mathcal{SH}(u_\Gamma) = \begin{cases} \mathcal{SH}^{(i)}(u_\Gamma) & \text{in } \Omega_i \\ u_\Gamma & \text{on } \Gamma \end{cases}$$

Our goal is to solve the linear system (3.6) by a preconditioned conjugated gradient method. This method does not require assembling the matrix S of the linear system, but only how to apply S to a vector w. By definition of S, applying S to a vector w is equivalent to applying matrices $K_{\Gamma\Gamma}$, $K_{I\Gamma}$, $K_{\Gamma I}$ and K_{II}^{-1} to subvectors of w. Among those applications, the K_{II}^{-1} is the most expensive one, however as we saw previously, this can be done in parallel.

3.5.2 BDD Preconditioning

Since all the remaining chapter is on the discrete space $\mathbf{X}_h \times M_h$ we simplify the notation by omitting subindex h.

Let us decompose the space $\mathbf{X} \times M = \left(\oplus_{i=1}^N \mathbf{X}_i \times M_i\right) \oplus (\mathbf{V}_\Gamma \times M_0)$ where $\mathbf{X}_i = \mathbf{X} \cap H_0^1(\Omega_i)$, $M_i = M \cap L_0^2(\Omega_i)$,

$$\mathbf{V}_\Gamma = \{\boldsymbol{v} \in \mathbf{X};\ \boldsymbol{v}|_{\Omega_i} = \mathcal{SH}^{(i)}(\boldsymbol{v}|_{\partial\Omega_i}),\ i = 1, \ldots, N\},$$

and

$$M_0 = \{q \in M;\ q|_{\Omega_i} = \text{const.},\ i = 1, \ldots, N\}.$$

We observe that the function $v \in \mathbf{V}_\Gamma$ is uniquely defined by its value on the interface Γ, so, any function defined on Γ can be uniquely identified to a function in $\mathbf{V}_\Gamma$.

Our objective now is to construct a parallel preconditioner M^{-1} for S in order to make the solution method scalable and well conditioned.

Following the balancing preconditioner for the Poisson equation, we look for a preconditioner of the form

$$M^{-1} = P_0 + (I - P_0) \sum_{i=1}^{N} R_i^T D_i^{-1} {S^{(i)}}^{-1} D_i^{-1} R_i (I - P_0), \tag{3.7}$$

where $S^{(i)}$ is the Schur complement of the local stiffness matrix $K^{(i)}$, the $R_i : \Gamma_h \to \Gamma_h^{(i)}$ is the discrete restriction operator, and the D_i^{-1} is a diagonal matrix defining a partition of unity on Γ_h, i.e., $\sum_{i=1}^{N} R_i^T D_i^{-1} R_i = I$ on Γ_h. The partition of unity may be defined through the *counting functions* defined for each subdomain as $\delta_i : \Gamma_h^{(i)} \to \mathbb{R}$ such that $\delta_i(x) =$ number of subdomains sharing the node $x \in \Gamma_h^{(i)}$, define D_i as $D_i = \text{diag}\{\delta_i\}$. When the problem has piecewise constant viscosity ν_i in each subdomain, and discontinuous across the interface Γ, then a better choice is to set

$$\delta_i = \frac{\sum_{j \in \mathcal{N}_x} \nu_j^{\gamma}(x)}{\nu_i^{\gamma}(x)}, \tag{3.8}$$

where $\gamma \in [1/2, \infty)$, and $\mathcal{N}_x$ is the set of indices of the subdomains that have the node x on their boundaries (see [45, 46]).

Remark 8. *The local problems ${S^{(i)}}^{-1}$ in (3.7) use natural boundary conditions*

$$\nu_i \nabla \boldsymbol{u} \cdot \boldsymbol{n} - pn = r \quad \text{on } \Gamma_h^{(i)}. \tag{3.9}$$

In this case the pressure is uniquely determined and therefore the pressure space are now taken on $L^2(\Omega)$.

When the boundary of a subdomain Ω_i does not intersect the boundary of the domain $\partial\Omega$, we have a *floating subdomain* Ω_i. Since the problem

$$S^{(i)} \begin{pmatrix} \boldsymbol{u}_\Gamma^{(i)} \\ p_0^{(i)} \end{pmatrix} = \begin{pmatrix} \widetilde{\boldsymbol{f}}_\Gamma^{(i)} \\ \widetilde{g}_0^{(i)} \end{pmatrix} \tag{3.10}$$

is equivalent as solving

$$\begin{pmatrix} K_{II}^{(i)} & K_{I\Gamma}^{(i)} \\ K_{\Gamma I}^{(i)} & K_{\Gamma\Gamma}^{(i)} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_I^{(i)} \\ p_I^{(i)} \\ \boldsymbol{u}_\Gamma^{(i)} \\ p_0^{(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \widetilde{\boldsymbol{f}}_\Gamma^{(i)} \\ \widetilde{g}_0^{(i)} \end{pmatrix},$$

then when Ω_i is a floating subdomain, $S^{(i)}$ has a kernel spanned by the *rigid body motions* (RBM) and therefore the linear system (3.10)

might not have a solution. In the two dimensional case the kernel basis is composed of three functions, two translations and one rotation. To avoid the issue of existence of solution, we introduce a coarse space $\mathbf{V}_0 \subset \mathbf{V}_\Gamma$ to enforce that when solving the linear system (3.10) the right hand side (RHS) is on the image of $S^{(i)}$, and since $S^{(i)}$ is symmetric, this is equivalent to have the RHS in $\mathrm{Ker}^{\perp}(S^{(i)})$. In addition we will require that the space $\mathbf{V}_0$ must be chosen so that the pairing $(\mathbf{V}_0,\ M_0)$ be stable, i.e., satisfies the inf-sup condition. We discuss possible choices of coarse spaces on Subsection 3.5.4.

3.5.3 Preconditioning in Matricial Form

Let $L_0 : \mathbf{V}_0 \to \Gamma$ be the matrix whose columns are the basis of the space $\mathbf{V}_0$. Then define the restriction operator

$$R_0 = \begin{pmatrix} L_0^T & 0 \\ 0 & I \end{pmatrix},$$

where I is the identity matrix of the size of the number of subdomains. To define a coarse problem Q_0, we set

$$S_0 = R_0 S R_0^T = \begin{pmatrix} L_0^T S_\Gamma L_0 & L_0^T B_0^T \\ B_0 L_0 & 0 \end{pmatrix},$$

and

$$Q_0 = R_0^T S_0^{-1} R_0.$$

The BDD preconditioner is then given by

$$M^{-1} = Q_0 + (I - Q_0 S) \sum_{i=1}^{N} Q_i (I - S Q_0),$$

and the preconditioned operator by

$$T = M^{-1} S = P_0 + (I - P_0) \sum_{i=1}^{N} P_i (I - P_0),$$

where $P_0 = Q_0 S$, $P_i = Q_i S$ and

$$Q_i = \begin{pmatrix} R_i^T D_i^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} S_\Gamma^{(i)} & {B_0^{(i)}}^T \\ B_0^{(i)} & 0 \end{pmatrix}^{-1} \begin{pmatrix} D_i^{-1} R_i & 0 \\ 0 & 0 \end{pmatrix}.$$

The minimal size coarse space $\mathbf{V}_0$ must be related to the local RBM associated to each subdomain Ω_i. Since the local problems are scaled

by D_i, we also scale the local RBM basis associated to Ω_i by D_i to define a coarse space so that the local problems (3.10) are compatible, i.e., for any $\boldsymbol{w} \in \mathbf{V}_\Gamma$

$$\left\langle \begin{pmatrix} D_i^{-1}R_i & 0 \\ 0 & * \end{pmatrix} S(I-P_0)\boldsymbol{w}, \boldsymbol{v}_i \right\rangle_{\Gamma_i} = 0 \quad \forall \boldsymbol{v}_i \in \mathbf{Ker}(S^{(i)}). \tag{3.11}$$

A desirable property of any parallel preconditioner is the scalability. To obtain that, the coarse space must also satisfies the following inf-sup condition

$$\sup_{\substack{\boldsymbol{v}_\Gamma \in \mathbf{V}_{\Gamma,h} \\ \boldsymbol{v}_\Gamma \neq 0}} \frac{(\nabla.\mathcal{SH}(\boldsymbol{v}_\Gamma), q_0)^2}{a(\mathcal{SH}\boldsymbol{v}_\Gamma, \mathcal{SH}\boldsymbol{v}_\Gamma)} \geq \beta_0 \|q_0\|^2_{L^2} \ \ \forall q_0 \in M_0. \tag{3.12}$$

We point out that the operator T is symmetric and positive definite with respect to S. The least eigenvalue of the preconditioned operator T in the S-norm is 1. By other side, when the inf-sup conditioin (3.12) holds, one can show the following bound in the S-norm for the condition of the preconditioned operator T holds (see [42]):

$$\mathrm{cond}_{S^{1/2}}(T) \leq \lambda_{\max} = \sup_{x \neq 0} \frac{< Tx, x >_S}{< x, x >_S} \leq C(1 + \frac{1}{\beta_0})\frac{1}{\beta^2}(1 + \log(\frac{H}{h}))^2 \tag{3.13}$$

where β is the inf-sup constant of the original problem (3.3) and C is a constant independent of H, h, β_0 and β.

3.5.4 The Coarse Space

The coarse space $\mathbf{V}_0$ plays an important role in the BDD preconditioning. This space must guarantee solvability for the local Neumann problems and scalability for the preconditioner. The minimum coarse space $\mathbf{V}_0$ for solvability is

$$\boldsymbol{V}_0^{(0)} = \begin{array}{l} \text{Rigid Body Motion of each subdomain } \Omega_i \text{ scaled by} \\ \text{diag}\{D_i^{-1}\} \text{ on } \Gamma_i \text{ and zero on the remaining nodes on } \Gamma. \end{array}$$

Thus, in the two dimensional case $\mathbf{V}_0^{(0)}$ has dimension $3\times$(number of subdomains). As we will see in the numerical results, the associated preconditioner T is not going to be scalable, therefore $\mathbf{V}_0^{(0)}$ must not satisfy the uniform inf-sup stability (3.12). This indicates that the coarse space should be enriched. Since our objective is unstructured

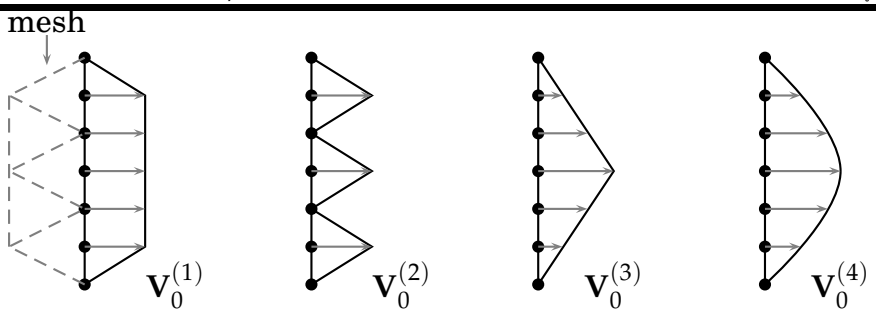


Figure 3.1: Sketch of the edge enrichment functions

mesh discretization, we need to design coarse space enrichments suitable for such discretizations. We enrich $\mathbf{V}_0^{(0)}$ with one coarse function per interface $\mathcal{E}_k$, i.e., connected components of an interface $\partial\Omega_i \cap \partial\Omega_j$.

Let $\mathcal{E}_k$ be an interface ordered by a sequence of vertices $(v_0, \ldots, v_{n_k})$ connected by fine edges on $\mathcal{T}_h(\partial\Omega_i \cap \partial\Omega_j)$. We define unit normal vectors $\boldsymbol{n}_j$ (for $j = 1, \ldots, (n_k - 1)$), by using the coordinates of v_j and its two neighboring vertices v_{j-1} and v_{j+1} on $\mathcal{T}_h(\partial\Omega_i \cap \partial\Omega_j)$. Let $\boldsymbol{\eta}_{j-1/2}$ and $l_{j-1/2}$ ($\boldsymbol{\eta}_{j+1/2}$ and $l_{j+1/2}$) be the unit normal and the length of the interval $[v_{j-1}, v_j]$ ($[v_j, v_{j+1}]$), respectively. Define

$$\boldsymbol{n}_j = (l_{j-1/2}\boldsymbol{\eta}_{j-1/2} + l_{j+1/2}\boldsymbol{\eta}_{j+1/2})/\|l_{j-1/2}\boldsymbol{\eta}_{j-1/2} + l_{j+1/2}\boldsymbol{\eta}_{j+1/2}\|_2.$$

To define the different coarse space enrichments we first define the weight functions w_k on each interface $\mathcal{E}_k$. We consider the following weight functions on $\mathcal{E}_k$ (see Fig. 3.1):

- for defining $\mathbf{V}_0^{(1)}$ let $w_k^{(1)} \equiv 1$
- for defining $\mathbf{V}_0^{(2)}$ let $w_k^{(2)}(v_j) = \begin{cases} 0 & \text{for } j \text{ even} \\ 1 & \text{for } j \text{ odd} \end{cases}$
- for defining $\mathbf{V}_0^{(3)}$ let $w_k^{(3)}(v_j) = \dfrac{\min\{d^1_{(j)},\, d^2_{(j)}\}}{\text{max\_dist}}$
- for defining $\mathbf{V}_0^{(4)}$ let $w_k^{(4)}(v_j) = \dfrac{d^1_{(j)} d^2_{(j)}}{(\text{max\_dist})^2}$, where $d^1_{(j)}$ and $d^2_{(j)}$ are defined as the l_2 distances to the boundary vertices v_0 and v_{n_k}, respectively, and let $\text{max\_dist} = \max_j\{d^1_{(j)},\, d^2_{(j)}\}$.

For each interface $\mathcal{E}_k$, we define the coarse function as

$$\boldsymbol{u}_k^{(r)}(v_j) = \begin{cases} w_k^{(r)}(v_j)\boldsymbol{n}_j & \text{for } j = 1, \ldots, (n_k - 1) \\ 0 & \text{for } j = 0,\ n_k \end{cases}$$

and then define the enriched coarse spaces $\mathbf{V}_0^{(r)}$, $r = 1, \ldots, 4$, as the space spanned by $\mathbf{V}_0^{(0)}$ and the coarse functions $\boldsymbol{U}_k^{(r)}$. The spaces $\mathbf{V}_0^{(1)}$ and $\mathbf{V}_0^{(2)}$ are quite easy to implement, even for the tridimensional case, since their implementation depend only on the normal vector at the vertices. Since the enrichment of $\mathbf{V}_0^{(1)}$ is already a basis of the RBM for structured meshes, we do not consider $\mathbf{V}_0^{(1)}$ on the numerical tests.

3.6 Implementation Aspects

In this section we discuss some of the implementation details of the code. A parallel software was developed in C using the PETSc library [3] for unstructured meshes. The unstructured meshes are generated using the 2D mesh generator EMC2 from INRIA [44]. The partitioning of the mesh is by elements and it is performed using the ParMETIS library [30].

3.6.1 BDD Implementation

To assemble the matrix B_0 and the right hand side g_0, we define a vector $e^{(i)}$ in order to recover the constant pressure function in the subdomain Ω_i; in the case of P_0 functions, $e^{(i)}$ is the vector of ones. The matrix $B_0^{(i)}$ is computed as ${B_0^{(i)}}^T = {B^{(i)}}^T e^{(i)}$, while the vector components of the vector g_0 are computed as $g_0^{(i)} = {e^{(i)}}^T g^{(i)}$. Since the discrete local pressure spaces are subspaces of $L_0^2(\Omega_i)$ and the global pressure space is a subspace of $L_0^2(\Omega)$, we employ Lagrange multipliers $\lambda^{(i)}$ to enforce zero average of each $p_l^{(i)}$ in Ω_i and another Lagrange multiplier μ to enforce zero average of p_0 in Ω.

To apply the BDD preconditioner it remains to deal with another issue when solving (3.10): the uniqueness of the Neumann solution for the floating subdomains. The natural way of dealing with such difficulty is to search for a solution $u_\Gamma^{(i)}$ which is orthogonal to the kernel of $S^{(i)}$, i.e., orthogonal to the local RBM. This is done by introducing three Lagrange multipliers per subdomain, i.e., one for each local RBM basis function.

3.6.2 A Higher Order Method

Having implemented the discretization with $\mathbf{P_2}/\mathrm{P}_0$ elements in PETSc we reuse all the *index sets* and *local to global mappings* defined for the $\mathbf{P_2}/\mathrm{P}_0$ elements to implement the $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ case. We add the bubble velocities and the linear average zero pressures on each element $K \in \mathcal{T}_h$, and then, through static condensation at the element level, we eliminate the bubble functions and the two average zero pressures. Let us denote by ψ_1, ψ_2, ψ_3 the usual P_1 basis functions associated to the element of reference $\widehat{K}$. To preserve the zero average in the pressure space $\mathrm{P}_1(\widehat{K})$, we map this basis into the new basis:

$$\begin{cases} \widetilde{\psi}_1 = 1 \\ \widetilde{\psi}_2 = x - 1/3 \\ \widetilde{\psi}_3 = y - 1/3. \end{cases}$$

Observe that $\widetilde{\psi}_2$ and $\widetilde{\psi}_3$ have zero average on $\widehat{K}$ while $\widetilde{\psi}_1$ is the basis for the P_0. At the element level we order the variables as

$$\begin{aligned} \boldsymbol{u}_p &\leftarrow \mathrm{P}_2 \text{ velocities} \\ p_c &\leftarrow \text{the constant pressure } \widetilde{\psi}_1 \\ \boldsymbol{u}_b &\leftarrow \text{bubble velocities} \\ p_l &\leftarrow \text{the linear functions } \widetilde{\psi}_2 \text{ and } \widetilde{\psi}_3, \end{aligned}$$

to obtain the following linear system

$$\begin{pmatrix} A_{pp} & B_{cp}^T & A_{pb} & B_{lp}^T \\ B_{cp} & 0 & B_{cb} & 0 \\ A_{bp} & B_{cb}^T & A_{bb} & B_{lb}^T \\ B_{lp} & 0 & B_{lb} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_p \\ p_c \\ \boldsymbol{u}_b \\ p_l \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_p \\ g_c \\ \boldsymbol{f}_b \\ g_l \end{pmatrix}.$$

As in the Remark 7, since A_{bb} is a positive definite matrix and B_{lb} is a full row rank matrix, the submatrix

$$\begin{pmatrix} A_{bb} & B_{lb}^T \\ B_{lb} & 0 \end{pmatrix}$$

is invertible and hence $\boldsymbol{u}_b$ and p_l can be eliminated. We note that the bubble functions vanish on the boundary of the elements, so $B_{cb} = B_{cb}^T = 0$, and therefore the constant function p_c can not be eliminated. The resulting matrix is then of the following form

$$\begin{pmatrix} \widetilde{A}_{pp} & B_{cp}^T \\ B_{cp} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_p \\ p_c \end{pmatrix} = \begin{pmatrix} \widetilde{\boldsymbol{f}}_p \\ \widetilde{g}_c \end{pmatrix}, \tag{3.14}$$

and after solving the linear system (3.14) we can recover the P_1 discontinuous pressure solution by solving the linear systems at the element level:

$$\begin{pmatrix} \boldsymbol{u}_b \\ p_l \end{pmatrix} = \begin{pmatrix} A_{bb} & B_{lb}^T \\ B_{lb} & 0 \end{pmatrix}^{-1} \left[\begin{pmatrix} \boldsymbol{f}_b \\ g_l \end{pmatrix} - \begin{pmatrix} A_{bp} & 0 \\ B_{lp} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_p \\ p_c \end{pmatrix} \right].$$

3.7 Numerical Results

A parallel software was developed in C using the PETSc library [3]. In order to study the scalability of the coarse space enrichments without the influence of the mesh partitioning, which may lead to irregular interface between subdomains, we consider in Subsections 3.7.1 and 3.7.2 a structured mesh in the domain $[0,1] \times [0,1]$ partitioned into $\sqrt{N} \times \sqrt{N}$ square subdomains. In Subsection 3.7.3 we consider an unstructured mesh example to study the parallel performance.

For the numerical experiments in Subsections 3.7.1 and 3.7.2 we impose Dirichlet boundary condition with the exact solution

$$\begin{cases} u_1(x,y) = x(1-x)\cos(x+y)\cos(x+3y) \\ u_2(x,y) = y(1-y)\sin(x+y)\sin(x+y) \\ p(x,y) = xy\exp(x+2y)\sin(x-y)\cos(y-x), \end{cases}$$

where we point out that $\nabla.\boldsymbol{u}$ is non-zero. Since the preconditioned operator T in (3.13) is symmetric positive definite with respect to S(see [42]), we use the preconditioned conjugated gradient (PCG) with the stopping criterion $\|r_k\|_2/\|r_0\|_2 \leq 10^{-6}$, where r_k is the residual at the iteration k. For solving the local problems we use the PETSc's LU with nested dissection reordering. The minimum eigenvalue is not presented in the tables since it is equal to one.

For the numerical experiments reported here we use a cluster of Linux PCs composed of 8 nodes, where each node has two Opteron processors and 8Gbytes of shared memory among it processors. Each processor is scored at 4.8Gflops. We remark that the code was compiled with debugging option, thus the timings can be at least twice faster if compiled with optimizations.

3.7.1 Constant Viscosity Tests

In this section all the numerical experiments are performed with a fixed viscosity $\nu = 1$ and using the discretization with $(\mathbf{P_2} + \mathbf{Bubble})/\mathrm{P}_1$

Table 3.1: The PCG iteration counts and the largest eigenvalues of the preconditioned operator T (within parenthesis) for different coarse spaces. We fix the local mesh to 32×32 and increase the number of subdomains.

	Coarse spaces			
Subdomains	$V_0^{(0)}$	$V_0^{(2)}$	$V_0^{(3)}$	$V_0^{(4)}$
3×3	19 (10.3)	19 (8.49)	17 (7.23)	16 (7.22)
4×4	23 (12.0)	22 (9.42)	20 (7.56)	20 (7.54)
5×5	27 (23.5)	25 (13.5)	20 (7.70)	20 (7.68)
6×6	28 (24.1)	24 (13.7)	20 (7.80)	20 (7.78)
7×7	30 (43.2)	26 (17.2)	20 (7.87)	20 (7.84)
8×8	35 (41.2)	27 (17.0)	21 (7.91)	20 (7.88)

Table 3.2: The PCG iteration counts and the largest eigenvalues of the preconditioned operator T (within parenthesis) for different coarse spaces. We fix the number of subdomains to 4×4 and refine the local meshes.

	Coarse spaces			
Local mesh	$V_0^{(0)}$	$V_0^{(2)}$	$V_0^{(3)}$	$V_0^{(4)}$
8×8	17 (7.87)	16 (4.72)	15 (4.30)	14 (4.27)
16×16	20 (9.83)	19 (6.80)	17 (5.82)	17 (5.79)
32×32	23 (12.0)	22 (9.42)	20 (7.56)	20 (5.74)

elements. On Table 3.1 we fix the mesh of the subdomains to 32×32 and increase the number of subdomains. On Table 3.2 we fix the number of subdomains to 4×4 and refine the mesh of the subdomains. These tables show the number of PCG iterations and the maximum eigenvalue (in parenthesis) for the different coarse spaces. We conclude from Table 3.1 that the coarse spaces $\mathbf{V}_0^{(0)}$ and $\mathbf{V}_0^{(2)}$ do not satisfy the uniform inf-sup stability (3.12), while the coarse spaces $\mathbf{V}_0^{(3)}$ and $\mathbf{V}_0^{(4)}$ provide scalable algorithms. From Table 3.2, we see that the performance of all the preconditioner does not depend much on the size of the local problems. This result is expected since the space $\mathbf{V}_0$ is large enough to balance all the local Neumann problems.

Table 3.3: The discretization errors of velocity for $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ and $\mathbf{P_2}/\mathrm{P}_0$ (within parenthesis). Fixing the number of subdomains to 4×4 and refining the local meshes.

	Discretization errors		
Local mesh	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_0$	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_1$	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{\mathrm{div}}$
4×4	3.64e-5 (5.73e-4)	3.88e-3 (3.63e-2)	2.82e-3 (3.31e-2)
8×8	3.71e-6 (1.47e-4)	9.13e-4 (1.84e-2)	6.93e-3 (1.69e-2)
16×16	4.13e-7 (3.73e-5)	2.18e-4 (9.26e-3)	1.71e-4 (8.52e-3)
32×32	4.97e-8 (9.40e-6)	5.39e-5 (4.64e-3)	4.27e-5 (4.27e-3)
64×64	6.60e-9 (2.36e-6)	1.34e-5 (2.33e-3)	1.07e-5 (2.14e-3)

For the subsequent numerical experiments we consider only the space $\mathbf{V}_0^{(4)}$ since Tables 3.1 and 3.2 show that this space is very effective.

On Tables 3.3 and 3.4 we compare the discrete errors of the $\mathbf{P_2}/\mathrm{P}_0$ (presented in parenthesis) and the $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ elements. We see that the $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ discretization is by far more accurate than the $\mathbf{P_2}/\mathrm{P}_0$. The convergence error rates for the $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ are 10, 4 and 4 for the velocity in the L^2, H^1, *div* norms, and 4 for the pressure in the L^2 norm, respectively. For the $\mathbf{P_2}/\mathrm{P}_0$ discretization the rates are 4, 2 and 2 for the velocity in the L^2, H^1, *div* norms, and 2 for the pressure in the L^2 norm, respectively.

Next we compare the discretizations $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ and $\mathbf{P_2}/\mathrm{P}_0$ with respect to execution and assembling times. We present in Table 3.5 the results for the $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ discretization and in parenthesis the results for the $\mathbf{P_2}/\mathrm{P}_0$ discretization. We compare the number of iterations, the maximum eigenvalue of the preconditioned system, the CPU time for assembling the stiffness matrix (T_1 in seconds), and the total CPU time (T_2) including the time for recovering the bubble velocity and the P_1 pressure. The Table 3.5 shows that the overall CPU time for the discretization $(\mathbf{P_2} + \mathbf{Bubble})/\mathrm{P}_1$ is not much larger than the $\mathbf{P_2}/\mathrm{P}_0$ one. Also we can see that the number of PCG iterations and the condition number are approximately the same for both discretizations.

Table 3.4: The L^2 discretization errors of pressure for $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ and $\mathbf{P_2}/\mathrm{P}_0$ (within parenthesis). Fixing the number of subdomains to 4×4 and refining the local meshes.

Local mesh	$\|p - p_h\|_0$
4×4	1.39e-2 (7.42e-2)
8×8	3.81e-3 (3.72e-2)
16×16	9.78e-4 (1.86e-2)
32×32	2.46e-4 (9.31e-3)
64×64	4.65e-5 (4.65e-3)

Table 3.5: PCG iteration counts (Its.), largest eigenvalue of the preconditioned operator T ($\lambda_{\max}$), CPU time for assembling the matrix and CPU times for all the running (T_2) for the discretizations $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ and $\mathbf{P_2}/\mathrm{P}_0$ (within parenthesis). Fixing the number of subdomains to 4×4 and refining the local meshes.

Local mesh	Its.	$\lambda_{\max}$	$T_1(s)$	$T_2(s)$
4×4	11 (13)	2.98 (3.42)	0.08 (0.06)	2.35 (2.30)
8×8	14 (14)	4.27 (4.57)	0.10 (0.07)	3.12 (2.90)
16×16	17 (16)	5.79 (5.96)	0.16 (0.10)	8.65 (8.53)
32×32	20 (18)	7.53 (7.61)	0.58 (0.34)	108.6 (107.1)
64×64	22 (21)	9.52 (9.51)	1.80 (0.93)	5687.1 (5682.6)

Table 3.6: PCG iteration counts and greatest eigenvalue (within parenthesis). Fixing the number of subdomains to 4×4.

γ	Local mesh	$\nu_2 = 10$	$\nu_2 = 100$	$\nu_2 = 1000$
$\gamma = 0.25$	8×8	19 (11.2)	25 (44.5)	26 (172)
	16×16	23 (16.0)	31 (65.3)	35 (254)
	32×32	25 (22.0)	35 (90.5)	43 (352)
$\gamma = 0.5$	8×8	15 (5.72)	17 (7.71)	17 (8.70)
	16×16	18 (7.93)	19 (10.7)	19 (12.1)
	32×32	20 (10.6)	22 (14.3)	22 (16.1)
$\gamma = 1$	8×8	13 (4.42)	11 (4.09)	11 (4.04)
	16×16	14 (5.72)	13 (5.13)	12 (5.04)
	32×32	16 (7.08)	15 (6.17)	13 (6.03)
$\gamma = 2$	8×8	13 (5.05)	11 (4.15)	11 (4.05)
	16×16	15 (6.57)	13 (5.21)	12 (5.05)
	32×32	17 (8.17)	15 (6.26)	13 (6.04)

3.7.2 Discontinuous Viscosities

In this section we assume that the viscosity is constant in each subdomain, however with a jump across the subdomains. We study the case where the viscosity is given by two constant values ν_1 and ν_2, in such a way that it has a checker-board pattern.

We consider the discretization $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ and fix $\nu_1 = 1$. In Table 3.6 we provide the number of iterations and the maximum eigenvalue (in parenthesis), for different values of the exponent γ; see (3.8). The best result is obtained when $\gamma = 1$, although for $\gamma > 1$ the condition numbers present similar behavior. In addition, as predicted in [45, 46], we confirm the strong deterioration on the performance of the algorithms when γ is less than 1/2 and ν_2 is large.

3.7.3 Parallel Performance

In order to analyze the parallel performance of the code we consider the discretization $(\mathbf{P_2} + \mathbf{Bubble})/\mathrm{P}_1$ and the coarse space enrichment $\mathbf{V}_0^{(4)}$ in the preconditioner. We also consider the domain Ω as in the Figure (3.2) with an unstructured mesh. We impose the following Dirichlet

boundary conditions

$$u(x,y) = \begin{cases} y(1-y); & \text{for } x = 0 \text{ (inflow)} \\ y(1-y); & \text{for } x = 6 \text{ (outflow)} \\ 0; & \text{otherwise (no-slip condition)} \end{cases}$$

In Table 3.7 we run problems with a mesh of 23008 elements (the system then have 116283 dofs). In order to study the scalability we solve a problem in one processor only with LU using nested dissection reordering. The speedup in N processors (S_N) is calculated as the ratio of total execution time in 1 processor (T_1) and in N processors (T_N) as $S_N = T_1/T_N$. The efficiency in N processors is computed as the ratio of the speedup in N processors and the number of processors, i.e., S_N/N. The CPU times show that the proposed preconditioner is more effective when the size of the local problems is small. This is due to the high cost of the local LU factorizations of the Dirichlet and Neumann matrices. The CPU time in assembling and in LU factorization of the coarse matrix is very small. The speedup factor grows super linearly when we increase the number of processors due to the smaller sizes of the local factorizations. The efficiency of the method grows due to the same reason; we point out that in the latter case of 32 subdomains there is an overload of the processors. We also mention that a postprocessing of the mesh partition can improve a little the iteration counts by smoothing the interface between the subdomains.

In Table 3.8 we fix the local mesh to 3222 elements. We point out that to setup the preconditioner for more than one subdomain it is required two LU factorizations, while in one subdomain we need just one. We remark that the band of the matrix in the one subdomain case is smaller than in the cases with more subdomains, due to the shape of the domain. Thus, the execution time for one subdomain is more than twice faster than the 4 subdomains case, however, from the case of 4 subdomains to 16 subdomains the increase in the execution time is almost all due to the iterative solver, that takes 15 more iterations than in the 4 subdomains case. Hence, by comparing the 4 and 16 subdomain cases, the scalability is obtained. We expect that the iteration counts will stabilize for large number of subdomains due to the theory and Table 3.1.

Figure 3.2: Domain for parallel performance test and sketch of an unstructured mesh.

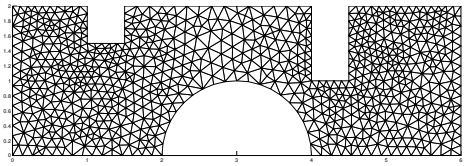


Table 3.7: This table shows the iteration counts (It.), total execution time (T_{tot}), the speedup factors, the efficiency (Eff.), and the CPU times to solve iteratively the linear system (T_{S}), to compute the LU factorizations of the local problems (T_{F}) and to compute the coarse matrix (this includes the LU factorization of the coarse matrix and is denoted T_{C}). The cases of 32 subdomains is performed by overloading some processors.

Subs.	Its.	T_{tot} (s)	Speedup	Eff.	T_{S}(s)	T_{F}(s)	T_{C}(s)
1 (LU)	–	4.91e+4	–	–	–	–	–
2	10	1.67e+4	2.94	1.47	1.06e+2	1.65e+4	1.15e+1
4	13	2.11e+3	23.3	5.82	3.85e+1	2.06e+3	5.83e+0
8	17	3.21e+2	153	19.1	2.17e+1	2.95e+2	3.49e+0
12	22	1.17e+2	420	35.0	2.56e+1	8.65e+1	2.52e+0
16	28	6.48e+1	758	47.4	2.09e+1	4.01e+1	1.84e+0
32	31	3.47e+1	1420	44.4	1.55e+1	1.13e+1	7.57e-1

Table 3.8: The local mesh is fixed to 3222 elements. The legends are the same as in the Table 3.7.

Subs.	Its.	T_{tot}(s)	T_{S}(s)	T_{F}(s)	T_{C}(s)
1 (LU)	–	1.41e+2	–	–	–
4	11	4.06e+2	1.32e+1	3.88e+2	2.80e+0
16	25	4.71e+2	5.43e+1	4.06e+2	5.38e+0

3.8 Conclusions

We propose four coarse spaces suitable for BDD preconditioning on unstructured meshes. We verify that the coarse spaces $\mathbf{V}_0^{(0)}$ and $\mathbf{V}_0^{(2)}$ are not stable, while the coarse spaces $\mathbf{V}_0^{(3)}$ and $\mathbf{V}_0^{(4)}$ are stable and scalable.

We introduce a new inf-sup stable discretization $(\mathbf{P_2} + \mathbf{Bubble})/\mathrm{P}_1$ which has been shown to be much more accurate than the $\mathbf{P_2}/\mathrm{P}_0$, without significant extra computational cost. This new discretization also has been shown to be scalable with the BDD preconditioner presenting similar iteration counts as the $\mathbf{P_2}/\mathrm{P}_0$ finite elements. As predicted in the theory developed in [45, 46] we have shown that the choice $\gamma \geq 1$ in the definition of the diagonal scaling (3.8) is a robust choice for highly discontinuous viscosities.

We develop a code based on PETSc library for 2D unstructured meshes, extensible to 3D meshes, with very impressive efficiency and speedup factors. In addition, as indicated by the numerical results, we can increase the performance of the local LU factorizations with the use of better reorderings.

CHAPTER 4

BDD Methods for Oseen type Equations

In this chapter we present the balancing domain decomposition methods for Oseen equations. Let us consider $\Omega \subset \mathbb{R}^d$ a polygonal domain. These equations are obtained by linearization of the non-linear term $\boldsymbol{u}.\nabla\boldsymbol{u}$ in the incompressible Navier-Stokes equations resulting in the system of partial differential equations:

$$\left\{\begin{array}{rcll} -2\nu\nabla.\varepsilon(\boldsymbol{u}) + \boldsymbol{a}.\nabla\boldsymbol{u} + \nabla p & = \boldsymbol{f} & \text{in } \Omega \\ -\nabla.\boldsymbol{u} & = g & \text{in } \Omega \\ \boldsymbol{u} & = \boldsymbol{u}_d & \text{on } \partial\Omega \end{array}\right. \tag{4.1}$$

where $\boldsymbol{a}$ is the advection vector field which we assume $\nabla.\boldsymbol{a} = 0$, and $\nu > 0$ is the kinematic viscosity. The Oseen equations mix the difficulties of the Stokes and the advection-diffusion equations.

We point out that as in the Stokes equations, the Dirichlet boundary condition $\boldsymbol{u}_d$ should satisfy the compatibility condition:

$$\int_\Omega g \, \mathrm{dx} = \int_{\partial\Omega} \boldsymbol{u}_d.\boldsymbol{n} \, \mathrm{ds}.$$

In order to simplify the presentation we assume zero Dirichlet boundary conditions, since the reduction to this case does not bring any extra difficulty.

The following relation is the analogous of Lemma 1 in a vectorial case and can be obtained similarly.

Lemma 9. *For all* $\boldsymbol{u}, \boldsymbol{v} \in H^1(\Omega)^d$ *and* $\boldsymbol{a} \in L^\infty(\Omega)^d$

$$\begin{aligned}(\boldsymbol{a}.\nabla\boldsymbol{u},\boldsymbol{v})_\Omega = \frac{1}{2}[(\boldsymbol{a}.\nabla\boldsymbol{u},\boldsymbol{v})_\Omega - (\boldsymbol{a}.\nabla\boldsymbol{v},\boldsymbol{u})_\Omega] \\ - \frac{1}{2}\int_\Omega (\nabla.\boldsymbol{a})\boldsymbol{u}.\boldsymbol{v}\,dx - \frac{1}{2}\int_{\partial\Omega}(\boldsymbol{a}.\boldsymbol{n})\boldsymbol{u}.\boldsymbol{v}\,dx\end{aligned} \tag{4.2}$$

Remark 10. *In the model problem we assume that* $\nabla.\boldsymbol{a} = 0$*, so, the third term in the right hand side of equation (4.2) vanishes. In the case that* $\boldsymbol{u}$ *or* $\boldsymbol{v}$ *be in* $H_0^1(\Omega)^d$ *the fourth term also vanishes.*

When dealing with balancing domain decomposition preconditioning we will need to solve local problems with natural boundary conditions, which for Oseen equations is different from Stokes equations. This natural boundary condition is:

$$\nu\nabla u_i.\boldsymbol{n} + \frac{1}{2}(\boldsymbol{a}.\boldsymbol{n})\boldsymbol{u}_i - pn_i = r_i \quad (i = 1, 2, \ldots, d)$$

where u_i is the i-th component of $\boldsymbol{u}$. The problem with this natural boundary condition has a unique pressure solution. In the case of a constant advection field the velocity solution is also unique.

4.1 The Variational Formulation

We restrict the presentation to the case $d = 2$, although it can be easily extended to higher dimensional cases. Defining $\boldsymbol{X} = H_0^1(\Omega)^2$ and $M = L_0^2(\Omega)$, the variational formulation of the Oseen equations is:

Find $\boldsymbol{u} \in \boldsymbol{X}$ *and* $p \in M$ *such that*

$$\begin{cases} \widetilde{a}(\boldsymbol{u},\boldsymbol{v}) + \widetilde{b}(\boldsymbol{v},p) &= F(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \boldsymbol{X} \\ \widetilde{b}(\boldsymbol{u},q) &= G(q) \quad \forall q \in M \end{cases} \tag{4.3}$$

where we define the bilinear forms

$$\widetilde{a}(\boldsymbol{u},\boldsymbol{v}) = 2\nu(\varepsilon(\boldsymbol{u}) : \varepsilon(\boldsymbol{v}))_\Omega + \frac{1}{2}\left[(\boldsymbol{a}.\nabla\boldsymbol{u},\boldsymbol{v})_\Omega - (\boldsymbol{a}.\nabla\boldsymbol{v},\boldsymbol{u})_\Omega\right]$$

$$\widetilde{b}(\boldsymbol{v},q) = -(\nabla.\boldsymbol{v},q)_\Omega$$

and the linear forms

$$F(\boldsymbol{v}) = (f,\boldsymbol{v})_\Omega$$

$$G(q) = -(g,q)_\Omega.$$

Remark 11. *The bilinear form*

$$a^*(\boldsymbol{u},\boldsymbol{v}) = 2\nu(\varepsilon(\boldsymbol{u}):\varepsilon(\boldsymbol{v}))_\Omega + (\boldsymbol{a}.\nabla\boldsymbol{u},\boldsymbol{v})_\Omega$$

is equivalent to the bilinear form $\widetilde{a}(\cdot,\cdot)$ when the problem has Dirichlet conditions prescribed on all the boundary $\partial\Omega$, since $\boldsymbol{v} \in H_0^1(\Omega)^2$.

To verify the existence and uniqueness of a solution, let us start defining the space $\boldsymbol{X}_{\mathrm{div},g} = \{\boldsymbol{v} \in H_0^1(\Omega); \quad \nabla.\boldsymbol{v} = g\}$ and observing that the variational problem (4.3) is equivalent to the following:

Find $\boldsymbol{u} \in \boldsymbol{X}_{div,g}$ such that

$$\widetilde{a}(\boldsymbol{u},\boldsymbol{v}) = F(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \boldsymbol{X}_{div,0} \tag{4.4}$$

For analysis it is suitable to decompose the bilinear form $\widetilde{a}(\cdot,\cdot)$ as

$$\widetilde{a}(\boldsymbol{u},\boldsymbol{v}) = a_{\mathrm{visc}}(\boldsymbol{u},\boldsymbol{v}) + a_{\mathrm{skew}}(\boldsymbol{u},\boldsymbol{v})$$

where

$$a_{\mathrm{visc}}(\boldsymbol{u},\boldsymbol{v}) = 2\nu(\varepsilon(\boldsymbol{u}):\varepsilon(\boldsymbol{v}))_\Omega$$

and

$$a_{\mathrm{skew}}(\boldsymbol{u},\boldsymbol{v}) = \frac{1}{2}\Big[(\boldsymbol{a}.\nabla\boldsymbol{u},\boldsymbol{v}) - (\boldsymbol{a}.\nabla\boldsymbol{v},\boldsymbol{u})\Big].$$

The continuity of the bilinear form $\widetilde{a}(\cdot,\cdot)$ can be proven by using the Cauchy-Schwarz inequality. The coercivity follows by applying Korn's inequality to the bilinear form $a_{\mathrm{visc}}(\cdot,\cdot)$ and by noting that $a_{\mathrm{skew}}(\cdot,\cdot)$ is skew-symmetric. This implies that the variational problem (4.4) has a unique solution.

4.2 Discretization

Let $\mathcal{T}_h$ a regular triangulation of the domain Ω. We consider the mixed finite element space with $\mathbf{P}_2/\mathrm{P}_0$ elements which was analyzed in the previous chapter and described in Section 3.4. As it was shown, this pairing of spaces satisfy the inf-sup stability condition.

The advection dominated problems are numerically unstable due to the advective part, as we pointed out in the first chapter. So, as in the advection-diffusion equation we also introduce a stabilization in the variational formulation. This stabilization also avoids the need of the discretization spaces to satisfies the inf-sup condition.

4.3 Stabilization of Advection Dominated Flows

We define the Reynolds number as $\mathbb{R}\mathrm{e} = \frac{\|\boldsymbol{a}\|_\infty L}{\nu}$ which is the quotient of inertia forces and stress forces, where L is the size of Ω. We say that a flow is *advective-dominated* when $\mathbb{R}\mathrm{e}$ is greater than one, and in other case to be *diffusive-dominated*.

It is well known that discretizing the variational problem (4.3) with finite elements even satisfying the inf-sup stability condition, will suffer oscillations on boundary layer and interior discontinuities in the advective-dominated case when the mesh in not sufficiently fine to resolve small scales. This problem can be avoided by introducing stabilizations in the variational formulation. The stabilizations we consider are the Galerkin least-squares and the Douglas-Wang (DW) variation of the GLS method. These stabilizations are presented and analysed in [19] with respect to convergence and stability. The variational formulation is: ***Find*** $(\boldsymbol{u}, p) \in X \times M$ ***such that***

$$B(\boldsymbol{u}, p; \boldsymbol{v}, q) = W(\boldsymbol{v}, q) \quad \forall (\boldsymbol{v}, q) \in X \times M$$

where the bilinear form $B(\cdot, \cdot; \cdot, \cdot)$ and the linear form $W(\boldsymbol{v}, q) = F(\boldsymbol{v}) + G(q)$ are defined in the GLS and in the DW stabilization methods as:

1. GLS:

$$B(\boldsymbol{u}, p; \boldsymbol{v}, q) = \widetilde{a}(\boldsymbol{u}, \boldsymbol{v}) + \widetilde{b}(\boldsymbol{v}, p) + \widetilde{b}(\boldsymbol{u}, q) + \sum_K \delta_K(\nabla.\boldsymbol{u}, \nabla.\boldsymbol{v}) + \sum_K \tau_K(-2\nu\nabla.\varepsilon(\boldsymbol{u}) + \boldsymbol{a}.\nabla\boldsymbol{u} + \nabla p, 2\nu\nabla.\varepsilon(\boldsymbol{v}) + a.\nabla\boldsymbol{v} - \nabla q)_{0,K}$$

$$F(\boldsymbol{v}) = (f, \boldsymbol{v}) + \sum_K \tau_K(f, 2\nu\nabla.\varepsilon(\boldsymbol{v}) + a.\nabla\boldsymbol{v})$$

$$G(q) = (g, q) + \sum_K \tau_K(f, -\nabla q)$$

2. DW:

$$B(\boldsymbol{u}, p; \boldsymbol{v}, q) = \widetilde{a}(\boldsymbol{u}, \boldsymbol{v}) + \widetilde{b}(\boldsymbol{v}, p) + \widetilde{b}(\boldsymbol{u}, q) + \sum_K \delta_K(\nabla.\boldsymbol{u}, \nabla.\boldsymbol{v}) + \sum_K \tau_K(-2\nu\nabla.\varepsilon(\boldsymbol{u}) + \boldsymbol{a}.\nabla\boldsymbol{u} + \nabla p, -2\nu\nabla.\varepsilon(\boldsymbol{v}) + a.\nabla\boldsymbol{v} - \nabla q)$$

$$F(\boldsymbol{v}) = (f, \boldsymbol{v}) + \sum_K \tau_K (f, -2\nu\nabla.\varepsilon(\boldsymbol{v}) + a.\nabla\boldsymbol{v})$$

$$G(q) = (g, q) + \sum_K \tau_K (f, -\nabla q)$$

We remark that these discretizations differ only in the sign of the term $2\nu\nabla.\varepsilon(v)$. In order to define the stabilization parameters δ_K and τ_K we introduce the elementwise Reynolds number

$$\mathbb{R}\mathrm{e}_K = \frac{m_k \|\boldsymbol{a}\|_\infty h}{4\nu}$$

which is mesh and discretization dependent. The term m_k is defined as

$$m_k = \min\{\frac{1}{3}, 2C_k\}$$

where C_k is the constant of the inverse inequality

$$C_k \sum_K h_K^2 \|\nabla.\varepsilon(\boldsymbol{v})\|_{0,K}^2 \leq \|\varepsilon(\boldsymbol{v})\|_0^2 \quad \forall \boldsymbol{v} \in V_h.$$

In each element $K \in \mathcal{T}_h$ we define the stabilization parameters δ_K and τ_K as

$$\delta_K = \|\boldsymbol{a}\|_{\infty,K} h \xi(\mathbb{R}\mathrm{e}_K)$$

and

$$\tau_K = \frac{h}{2\|\boldsymbol{a}\|_{\infty,K}} \xi(\mathbb{R}\mathrm{e}_K)$$

where

$$\xi(\mathbb{R}\mathrm{e}_K) = \begin{cases} \mathbb{R}\mathrm{e}_K; & 0 \leq \mathbb{R}\mathrm{e}_K < 1 \\ 1; & \mathbb{R}\mathrm{e}_K \geq 1. \end{cases}$$

In [19], Franca and Frey show that the GLS stabilization is unstable for large values of τ_K while the Douglas-Wang variation is more stable in terms of the choice of the stabilization parameter τ_K. Thus, we choose the DW stabilization for our stabilized variational discrete model:

Find $(\boldsymbol{u}, p) \in \boldsymbol{X} \times M$ ***such that***

$$\begin{cases} a(\boldsymbol{u}, \boldsymbol{v}) + b^{\mathrm{p}}(\boldsymbol{v}, p) = F(\boldsymbol{v}) & \forall \boldsymbol{v} \in X \\ b^{\mathrm{u}}(\boldsymbol{u}, q) + c(p, q) = G(q) & \forall q \in M \end{cases}$$

where the bilinear forms are given as

$$a(\boldsymbol{u},\boldsymbol{v}) = B_{\mathrm{DW}}(\boldsymbol{u},0;\boldsymbol{v},0),$$
$$b^{\mathrm{p}}(\boldsymbol{v},p) = B_{\mathrm{DW}}(0,p;\boldsymbol{v},0),$$
$$b^{\mathrm{u}}(\boldsymbol{u},q) = B_{\mathrm{DW}}(\boldsymbol{u},0;0,q)$$

and

$$c(p,q) = B_{\mathrm{DW}}(0,p;0,q).$$

Therefore, the linear system obtained from this discretization is

$$\begin{pmatrix} A & {B^{\mathrm{p}}}^T \\ B^{\mathrm{u}} & C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{4.5}$$

Denoting the matrix B_{stab} as the discretization of the term

$$\sum_K \tau_K(\nabla p, -2\nu\nabla.\varepsilon(\boldsymbol{v}) + a.\nabla\boldsymbol{v})$$

and by B_{oseen} the discretization of the term $b(v,p)$, we obtain that $B^{\mathrm{p}} = B_{\mathrm{oseen}} + B_{\mathrm{stab}}$ and $B^{\mathrm{u}} = B_{\mathrm{oseen}} - B_{\mathrm{stab}}$.

4.4 BDD Method for Oseen Equations

In this section we present the balancing domain decomposition method for Oseen equations. This is an extension of the BDD method for Stokes equations, presented in the previous chapter.

As before, we decompose the domain Ω into N non-overlapping subdomains Ω_i and the interface $\Gamma = \cup_{i=1}^{N}\partial\Omega_i \setminus \partial\Omega$, so that

$$\Omega = \bigcup_{i=1}^{N} \Omega_i \cup \Gamma.$$

In addition we denote by Γ_i the nodes on $\partial\Omega_i \setminus \partial\Omega$.

4.4.1 Substructuring in Matrix Form

Similarly to the Stokes case, we reorder and denote the variables as follows:

- $\boldsymbol{u}_I$ $\leftarrow$ interior velocities
- p_I $\leftarrow$ pressures with zero average in each subdomain Ω_i
- $\boldsymbol{u}_\Gamma$ $\leftarrow$ interface velocities
- p_0 $\leftarrow$ constant pressure in each Ω_i and with zero average in Ω

As it is usual in substructuring methods we eliminate some variables of the linear system to obtain a condensated system. In balancing domain decomposition we eliminate the interior degrees of freedom, resulting in a linear system in the complementary degrees of freedom of the velocity and the pressure variables with mean value zero in each substructure, by now, $\boldsymbol{u}_\Gamma$ and p_0 .

Using this reordering, the matrix of the discrete system (4.5) can be written as:

$$K = \begin{pmatrix} K_{II} & K_{I\Gamma} \\ K_{\Gamma I} & K_{\Gamma\Gamma} \end{pmatrix} = \begin{pmatrix} A_{II} & B_{II}^{\mathrm{p}^T} & A_{I\Gamma} & B_{0I}^{\mathrm{p}^T} \\ B_{II}^{\mathrm{u}} & C_{II} & B_{I\Gamma}^{\mathrm{u}} & C_{I\Gamma} \\ A_{I\Gamma} & B_{I\Gamma}^{\mathrm{p}^T} & A_{\Gamma\Gamma} & B_0^{\mathrm{p}^T} \\ B_{0I}^{\mathrm{u}} & C_{\Gamma I} & B_0^{\mathrm{u}} & C_{\Gamma\Gamma} \end{pmatrix}.$$

Since $\nabla.p_0 = 0$ and $\int_\Omega \nabla.\boldsymbol{u}_I \, \mathrm{dx} = 0$, then the submatrices B_{0I}, $C_{\Gamma I}$, $C_{I\Gamma}$ and $C_{\Gamma\Gamma}$ are all null. On the other hand, the stabilization part of B_0^{u} and B_0^{p} are also null, therefore $B_0 = B_0^{\mathrm{u}} = B_0^{\mathrm{p}}$. Then, even though we are introducing a stabilization in the variational model, the matricial form of the system is analogous to the Stokes one.

Eliminating the interior variables $\boldsymbol{u}_I$ and p_I by static condensation we obtain the system

$$S \begin{pmatrix} \boldsymbol{u}_\Gamma \\ p_0 \end{pmatrix} = \begin{pmatrix} \widetilde{f}_\Gamma \\ \widetilde{g}_0 \end{pmatrix} \tag{4.6}$$

where

$$S = K_{\Gamma\Gamma} - K_{\Gamma I} K_{II}^{-1} K_{I\Gamma} = \begin{pmatrix} S_\Gamma & B_0^T \\ B_0 & 0 \end{pmatrix},$$

and

$$\begin{pmatrix} \widetilde{f}_\Gamma \\ \widetilde{g}_0 \end{pmatrix} = \begin{pmatrix} f_\Gamma \\ g_0 \end{pmatrix} - K_{\Gamma I} K_{II}^{-1} \begin{pmatrix} f_I \\ g_I \end{pmatrix}.$$

Remark 12. *Since A_{II} is positive definite (by Korn's inequality) and B_{II} has full row rank, the matrix K_{II} is invertible. We note also that since B_{0I} is null, it is not possible to eliminate p_0.*

Once we have solved the linear system (4.6), we can obtain the solutions $\boldsymbol{u}_I$ and p_I by solving

$$\begin{pmatrix} \boldsymbol{u}_I \\ p_I \end{pmatrix} = \begin{pmatrix} A_{II} & B_{II}^{\mathrm{p}^T} \\ B_{II}^{\mathrm{u}} & 0 \end{pmatrix}^{-1} \left[\begin{pmatrix} f_I \\ g_I \end{pmatrix} - \begin{pmatrix} A_{I\Gamma} & 0 \\ B_{I\Gamma}^{\mathrm{u}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_\Gamma \\ p_0 \end{pmatrix} \right],$$

where we observe that $\boldsymbol{u}_I$ and p_I do not depend on p_0. As in the earlier chapters, if we reorder the interior variables by subdomain we obtain

$$K_{II} = \begin{pmatrix} K_{II}^{(1)} & & 0 \\ & \ddots & \\ 0 & & K_{II}^{(N)} \end{pmatrix}.$$

Again, as in the Stokes case these matrices are decoupled and the computation of the multiplication of K_{II}^{-1} by a vector can be performed in parallel. The local Oseen harmonic extension is defined as

$$\mathcal{OH}^{(i)}(u_{\Gamma_i}) = \begin{pmatrix} -K_{II}^{-1}K_{I\Gamma}u_\Gamma \\ u_{\Gamma_i} \end{pmatrix}.$$

This is equivalent to solving the variational problem with Dirichlet boundary condition

$$\begin{cases} a(u,v) + b^{\mathrm{p}}(v,p) = F(v) \\ b^{\mathrm{u}}(u,q) + c(p,q) = G(q) \\ u|_\Gamma = u_\Gamma \end{cases}$$

where we point out that the term $c(p,q)$ vanishes since the pressures are constant on each element.

Analogously, we define the Oseen harmonic extension as

$$\mathcal{OH}(u_\Gamma) = \begin{cases} \mathcal{OH}^{(i)}(u_\Gamma) & \text{in } \Omega_i \\ u_\Gamma & \text{on } \Gamma \end{cases}$$

4.4.2 BDD preconditioning for Oseen

Since in the remaining of the chapter we only use the discrete space $X_h \times M_h$ we simplify the notation by omitting the subindex h.

Similarly as in the Stokes problem we decompose the space $X \times M$ as

$$\boldsymbol{X} \times M = (\oplus_i \boldsymbol{X}_i \times M_i) \oplus (\boldsymbol{V}_\Gamma \times M_0)$$

where $\mathbf{X}_i = \mathbf{X} \cap H_0^1(\Omega_i)$, $M_i = M \cap L_0^2(\Omega_i)$,

$$\mathbf{V}_\Gamma = \{\boldsymbol{v} \in \mathbf{X};\ \boldsymbol{v}|_{\Omega_i} = \mathcal{OH}^{(i)}(\boldsymbol{v}|_{\partial\Omega_i}),\ i = 1, \ldots, N\},$$

and

$$M_0 = \{q \in M;\ q|_{\Omega_i} = \text{const.},\ i = 1, \ldots, N\}.$$

Since the local Oseen harmonic extensions $\mathcal{OH}^{(i)}$ are uniquely defined by the values on $\partial\Omega_i$, the functions $v \in V_\Gamma$ are also uniquely defined by its value on Γ.

The balancing preconditioner for the Schur complement system of the stabilized Oseen linear system is of the form

$$M^{-1} = P_0 + (I - P_0) \sum_{i=1}^{N} R_i^T D^{-1} S_i^{-1} D_i^{-1} R_i (I - P_0)$$

where S_i^{-1} is the Schur complement of the local stiffness matrix K_i^{-1}, P_0 is a projection on a coarse space V_0 to be defined in the following sections, and P_i is a projection like operator analogous as the Stokes one, where we need to solve local problems

$$S^{(i)} \begin{pmatrix} \boldsymbol{u}_\Gamma^{(i)} \\ p_0^{(i)} \end{pmatrix} = \begin{pmatrix} \widetilde{\boldsymbol{f}}_\Gamma^{(i)} \\ \widetilde{g}_0^{(i)} \end{pmatrix}, \tag{4.7}$$

which is equivalent to solve the natural boundary condition problem

$$\begin{pmatrix} K_{II}^{(i)} & K_{I\Gamma}^{(i)} \\ K_{\Gamma I}^{(i)} & K_{\Gamma\Gamma}^{(i)} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_I^{(i)} \\ p_I^{(i)} \\ \boldsymbol{u}_\Gamma^{(i)} \\ p_0^{(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \widetilde{\boldsymbol{f}}_\Gamma^{(i)} \\ \widetilde{g}_0^{(i)} \end{pmatrix}.$$

We recall that in the Stokes balancing, the local problems of the floating subdomains have non-trivial kernel, spanned by the *rigid body motions* (RBM). The kernel of the local Oseen problems in the floating subdomains, as well as in the advection-diffusion equation, is contained in the kernel of the Stokes problem, that is, in the set of RBM functions.

4.4.3 Preconditioner in Matricial Form

Suppose we have defined the coarse subspace $V_0 \subset V_\Gamma$, and let $L_0 : V_0 \to \Gamma$ be the extension matrix whose columns are the basis of the coarse space. Then we define the restriction operator

$$R_0 = \begin{pmatrix} L_0^T & 0 \\ 0 & I \end{pmatrix}$$

where I is the identity matrix whose size is the number of subdomains. To define the coarse problem Q_0 we set

$$S_0 = R_0 S R_0^T = \begin{pmatrix} L_0^T S_\Gamma L_0 & L_0^T B_0^T \\ B_0 L_0 & 0 \end{pmatrix}$$

and

$$Q_0 = R_0^T S_0^{-1} R_0.$$

The preconditioner is defined as

$$M^{-1} = Q_0 + (I - Q_0 S) \sum_{i=1}^{N} Q_i (I - S Q_0),$$

and the preconditioned operator by

$$T = M^{-1} S = P_0 + (I - P_0) \sum_{i=1}^{N} P_i (I - P_0),$$

where $P_0 = Q_0 S$, $P_i = Q_i S$ and

$$Q_i = \begin{pmatrix} R_i^T D_i^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} S_\Gamma^{(i)} & B_0^{(i)^T} \\ B_0^{(i)} & 0 \end{pmatrix}^{-1} \begin{pmatrix} D_i^{-1} R_i & 0 \\ 0 & 0 \end{pmatrix}.$$

The minimal size coarse space $\mathbf{V}_0$ must be related to the local RBM associated to each subdomain Ω_i. Since the local problems are scaled by D_i, we also scale the local RBM basis associated to Ω_i by D_i to define a coarse space so that the local problems (4.7) are compatible, i.e., for any $\boldsymbol{w} \in \mathbf{V}_\Gamma$

$$\left\langle \begin{pmatrix} D_i^{-1} R_i & 0 \\ 0 & * \end{pmatrix} S(I - P_0)\boldsymbol{w}, \boldsymbol{v}_i \right\rangle_{\Gamma_i} = 0 \quad \forall \boldsymbol{v}_i \in \mathbf{Ker}(S^{(i)^T}). \tag{4.8}$$

4.4.4 Coarse Space

We consider the coarse space as the rigid body motions (RBM) scaled by $\mathrm{diag}\{D_i^{-1}\}$ enriched with quadratic functions on each coarse edge $\partial\Omega_i \cap \partial\Omega_j$ on the coarse grid. We choose this space since this shows the better convergence and stability results in the Stokes case; see Section 3.7.

We remark that this choice will also introduce compatibility in the local problems in the case of the existence of a kernel generated by the RBM functions, which is the unique possible kernel of the Oseen equations.

4.5 Numerical Results

For numerical tests let us consider the domain $\Omega = [0,1] \times [0,1]$ partitioned into $\sqrt{N} \times \sqrt{N}$ square subdomains as sketched in Figure 2.1. Since the linear system is non-symmetric we consider the GMRES method for the iterative solver. The convergence is achieved when the relative residual $\|M^{-1}r_n\|_0/\|M^{-1}r_0\|_0$ is less than $1.e-6$.

The tests are performed using as exact solution

$$\begin{cases} u_1(x,y) = \exp(\cos(x) - \sin(y)).\sin(3x+1) \\ u_2(x,y) = \exp(2x^2 + \sin(y)).(\cos(2x-y) + \sin(2y-x)) \\ p = \exp(2x+y) \end{cases}$$

by imposing Dirichlet boundary condition. We consider in the numerical experiments a constant advection field $\boldsymbol{a} = (1,3)$.

In this section we present the numerical results of the implementation of the Oseen problem.

On Table 4.1 we fix the global mesh to 128×128 and analyze, for different Reynolds numbers, the stability of the preconditioner on different choices of local mesh size and number of subdomains. The iterations show stability of the preconditioner for all the Reynolds numbers we considered. On the other hand, the iterations are dependent on the Reynolds number.

On Table 4.2 we fix the local mesh to 4×4, 8×8 and 16×16, the viscosity to $\nu = 0.00002$ (so, $\mathbb{R}\mathrm{e} = 1.58e+5$) and increase the number of subdomains. The number of iterations necessary for convergence seems to increase sub-linearly as we increase the number of subdomains. Hence, this table shows an almost scalability of the preconditioner.

Analysing the case of a fixed number of subdomains to 8×8 on Table 4.2 we note that the iterations of the preconditioned GMRES seems to increase sub-linearly, and so, the preconditioner is more dependent on the local mesh than the number of subdomains.

4.6 Conclusions

We extend the BDD preconditioner for Stokes equations to the stabilized Oseen equations, using edge enrichment for the coarse space. We show that the kernel of the local problems is a subspace of the kernel

Table 4.1: Preconditioned GMRES iteration counts. The global mesh is fixed to 128×128, and we change the number of subdomains and the local meshes (within parenthesis). The constant advection field is $(1, 3)$.

	Subdomains (local mesh)			
$\mathbb{R}e_K$ (viscosity)	32 (4)	16 (8)	8 (16)	4(32)
1.45e+5 (2.e-8)	79	72	65	39
1.45e+4 (2.e-7)	79	72	65	39
1.45e+3 (2.e-6)	78	72	65	39
145 (2.e-5)	77	71	63	39
14.5 (2.e-4)	63	62	57	37
1.45 (2.e-3)	27	42	46	33
0.145 (2.e-2)	21	22	29	27

Table 4.2: Preconditioned GMRES iteration counts and Reynods number within parenthesis. We fix the local mesh to $H/h = 4, 8, 16$. We also fix the viscosity to 0.00002 and the advection field to $(1, 3)$. The local Reynolds number ($\mathbb{R}e_K$) is presented within parenthesis.

	Local mesh (H/h)		
Subdomains	4	8	16
3×3	18 (1.6e+3)	20 (7.8e+2)	24 (3.9e+2)
4×4	23 (1.2e+3)	28 (5.8e+2)	33 (2.9e+2)
8×8	39 (5.8e+2)	48 (2.9e+2)	63 (1.5e+2)
12×12	47 (3.9e+2)	60 (1.9e+2)	81 (9.7e+1)
16×16	53 (2.9e+2)	71 (1.5e+2)	95 (7.3e+1)

of the local problems associated to the equivalent Stokes problem (that is, the problem withou the advective term).

The scalability of the preconditioner for Oseen equations is shown to be achevied only in the cases when the local Péclet number is order of one. Also the local solvers and the coarse solvers are somehow sensitive to the Péclet number when it becomes greater than one. We believe that in fact they are sensitive to the Péclet number even when it is less than one, however without significant influence in the iterations.

CHAPTER 5

Future Works

For future work, we plan to

- develop an enrichment of the coarse space based on stabilized finite element technique in order to better deal with the non-symmetric part of the coarse and local problems, and the corresponding analysis of the methods;
- extend the preconditioner to the advection-diffusion and the Oseen problems with discontinuous coeficients;
- study the behaviour of the preconditioner for advection-diffusion and Oseen problems on rotating or complicated flow fields;
- extend the BDDC algorithm for the Oseen problems;
- develop a coarse space enrichment for unstructured meshes for Stokes and Oseen problems that makes the setting of the coarse space enrichment easier for implementation of the preconditioner;
- increase performance of the PETSc code by introducing better reorderings for LU factorization; and
- develop a preconditioner to the core-flow model, i.e., introducing the interface equation.

Bibliography

[1] Y. Achdou, P. Le Tallec, F. Nataf, and M. Vidrascu. A domain decomposition preconditioner for an advection-diffusion problem. *Comput. Methods Appl. Mech. Engrg.*, 184:145–170, 2000.

[2] R. Bai, K. Kelkar, and D.D. Joseph. Direct simulation of interfacial waves in a high-viscosity-ratio and axisymmetric core-annular flow. *J. Fluid Mech.*, 327:1–34, 1996.

[3] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang. PETSc web page : http://www.mcs.anl.gov/petsc, 2001.

[4] A.C. Bannwart and J.W.V. Prada. Production of heavy oil via core flow. In paper SPE 53688, editor, *6th Latin American and Caribbean Petroleum Engineering Conference - LACPEC'99*, Caracas, Venezuela, 1999.

[5] A.C. Bannwart and J.W.V. Prada. Modeling of vertical core annular flows and application to heavy oil production. In *Proceedings of the ETCE2000 and OMAE 2000 Joint Conference - Energy for the New Millenium, CDROM*, New Orleans, LA, February14-17 2000.

[6] D. Braess. *Finite Elements: Theory, Fast Solvers and Applications to Solid Mechanics*. Cambridge University Press, 2001.

[7] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin methods for advection dominated flows. In *Third In-*

ternational Conference on Finite Element Methods in Fluid Flows, Banf, Canada, 1980.

[8] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convective dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32:199–259, 1982.

[9] X.-C. Cai, W.D. Gropp, and D.E. Keyes. A comparison of some domain decomposition algorithms for nonsymmetric elliptic problems. In D.E. Keyes, T.F. Chan, G.A. Meurant, J.S. Scroggs, and R.G. Voigt, editors, *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 224–235, Philadelphia, PA, 1992. SIAM.

[10] X.-C. Cai and O. Widlund. Domain decomposition algorithms for indefinite elliptic problems. *SIAM J. Sci. Statist. Comput.*, 13(1):243–258, 1992.

[11] X.-C. Cai and O. Widlund. Multiplicative Schwarz algorithms for some nonsymmetric and indefinite problems. *SIAM J. Numer. Anal.*, 30(4):936–952, 1993.

[12] C.R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, 25(1):246–258, 2003.

[13] J. Douglas and J. Wang. An absolutely stabilized finite element method for the Stokes problem. *Math. Comp.*, 52:495–508, 1989.

[14] M. Dryja and O. Widlund. Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. *Comm. Pure Appl. Math.*, 48(2):121–155, 1995.

[15] H.C. Elman, D.J. Silvester, and A.J. Wathen. Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations. *Numer. Math.*, pages 665–688, 2002.

[16] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. 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.

[17] C. Farhat, J. Mandel, and F.-X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Comput. Methods Appl. Mech. Engrg.*, 115(3-4):365–385, 1994.

[18] C. Farhat and F.-X. Roux. A method of Finite Element Tearing and Interconnecting and its parallel solution algorithm. *Int. J. Numer. Meth. Eng.*, 32:1205–1227, 1991.

[19] L.P. Franca and S.L. Frey. Stabilized finite element methods: II. the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 99:209–233, 1992.

[20] L.P. Franca, S.L. Frey, and T.J.R. Hughes. Stabilized finite element methods: I. Application to the advective-diffusive model. *Comput. Methods Appl. Mech. Engrg.*, 95:253–276, 1992.

[21] L.P. Franca, T.J.R. Hughes, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska-Brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Comput. Methods Appl. Mech. Engrg.*, 59:85–99, 1986.

[22] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations: Theory and algorithms*. Springer Series in Computational Mathematics. Springer, Berlin, 1986.

[23] P. Goldfeld, L.F. Pavarino, and O. Widlund. Balancing Neumann-Neumann preconditioners for mixed approximations of heterogeneous problems in linear elasticity. *Numer. Math.*, 95(2):283–324, 2003.

[24] I. Harari and T.J.R. Hughes. What are C and h? Inequalities for the analysis and design of finite element methods. *Comput. Methods Appl. Mech. Engrg.*, 97:157–192, 1992.

[25] T.J.R. Hughes and A.N. Brooks. A multidimensional upwind scheme with no crosswind diffusion. In T.J.R. Hughes, editor, *Finite Element Methods for Convection Dominated Flows*, pages 19–35, New York, 1979. ASME.

[26] T.J.R. Hughes and A.N. Brooks. A theoretical framework for Petrov-Galerkin methods with discontinuous weighting functions:

Application to the streamline upwind procedure. In R.H. Gallagher, G.F. Carey, J.T. Oden, and O.C. Zienkiewicz, editors, *Finite Elements in Fluids IV*, pages 46–65. Wiley, Chichester, 1982.

[27] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, 1987.

[28] D.D. Joseph, R. Bai, K.P. Chen, and Y.Y. Renardy. Core-annular flows. *Annu. Rev. Fluid Mech.*, 29:65–90, 1997.

[29] D.D. Joseph and Y.Y. Renardy. *Fundamental of Two Fluid Dynamics. Part 2: Lubricated Transport, Drops and Miscible Liquids,*. Interdisciplinary Applied Mathematics Series vol. 4. Springer-Verlag New York, 1993.

[30] G. Karypis, K. Schloegel, and V. Kumar. Parmetis – parallel graph partitioning and sparse matrix ordering library. version 3.1. Web page: http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview.

[31] D. Kay, D. Loghin, and A. Wathen. A preconditioner for the steady-state Navier-Stokes equations. *SIAM J. Sci. Comput.*, 24(1):237–256, 2002.

[32] A. Klawonn and O. Widlund. FETI and Neumann-Neumann iterative substructuring methods: connections and new results. *Comm. Pure Appl. Math*, 54(1):57–90, 2001.

[33] A. Klawonn and O. Widlund. FETI-DP methods for elliptic problems with discontinuous coefficients in three dimensions. In *Domain Decomposition Methods in Science and Engineering (Lyon, 2000)*, Theory Eng. Appl. Comput. Methods, pages 405–411, Barcelona, 2002. Internat. Center Numer. Methods Eng. (CIMNE).

[34] A. Klawonn, O.B. Widlund, and M. Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM J. Numer. Anal.*, 40(1):159–179, 2002.

[35] J. Li and Y. Renardy. Direct simulation of unsteady axisymmetric core-annular flow with high viscosity ratio. *J. Fluid Mech.*, 391:123–149, 1999.

[36] J. Li and O. Widlund. FETI-DP, BDDC, and block Cholesky methods. Technical Report TR2004-857, Courant Institute of Mathematical Sciences, December 2004. Department of Computer Science, New York University.

[37] J. Li and O. Widlund. BDDC algorithms for incompressible Stokes equations. Technical Report TR2005-861, Courant Institute of Mathematical Sciences, April 2005. Department of Computer Science, New York University.

[38] G. Lube and A. Auge. Regularized mixed finite element approximations of incompressible flow problems. II. Navier-Stokes flow. Preprint 15/91, Otto von Guericke University, 1991.

[39] J. Mandel. Balancing domain decomposition. *Comm. Numer. Methods Engrg.*, 9(3):233–241, 1993.

[40] J. Mandel and M. Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comp.*, 65:1387–1401, 1996.

[41] J. Mandel and C.R. Dohrmann. Convergence of balancing domain decomposition by constraints and energy minimization. *Numer. Linear Algebra Appl.*, 10(7):639–659, 2003.

[42] L.F. Pavarino and O. Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. *Comm. Pure Appl. Math.*, 55(3):302–335, 2002.

[43] Y.-H. De Roeck and P. Le Tallec. Analysis and test of a local domain decomposition preconditioner. In R. Glowinsky, Y. Kuznetsov, G. Meurant, J. Périaux, and O. Widlund, editors, *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1991. SIAM.

[44] E. Saltel and F. Hecht. EMC2 Wysiwyg 2d finite elements mesh generator. INRIA. EMC2 web page: http://www-rocq1.inria.fr/gamma/cdrom/www/emc2/en.

[45] M. Sarkis. Two-level Schwarz methods for nonconforming finite elements and discontinuous coefficients. In N. Duane Melson,

Thomas A. Manteuffel, and Steve F. McCormick, editors, *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods, Volume 2*, volume 3224, pages 543–566, Hampton VA, 1993. NASA.

[46] M. Sarkis. Nonstandard coarse spaces and Schwarz methods for elliptic problems with discontinuous coefficients using non-conforming element. *Num. Math.*, 77:383–406, 1997.

[47] B.F. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, Cambridge, 1996.

[48] R. Stenberg. Analysis of mixed finite elements methods for the Stokes problem: a unified approach. *Math. Comp.*, 42(164):9–23, 1984.

[49] R. Stenberg. Error analysis of some finite element methods for the Stokes problem. *Math. Comp.*, 54(190):495–508, 1990.

[50] P. Le Tallec, J. Mandel, and M. Vidrascu. A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems. *SIAM J. Numer. Anal.*, 35:836–867, 1998.

[51] A. Toselli. FETI domain decomposition methods for scalar advection-diffusion problem. Technical Report TR2000-800, Courant Institute of Mathematicla Sciences, Department of Computer Science, New York University, 2000.

[52] A. Toselli and O. Widlund. *Domain Decomposition Methods: Algorithms and Theory.*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.