Robust Two-Level Lower-Order Preconditioners for a Higher-Order Stokes Discretization with Highly Discontinuous Viscosities

Duilio Conceição[1]*, Paulo Goldfeld[2], and Marcus Sarkis[1]

[1] IMPA, Rio de Janeiro, Brazil,
dtadeu@fluid.impa.br, msarkis@fluid.impa.br
[2] UFRJ, Instituto de Matemática, Rio de Janeiro, Brazil,
goldfeld@ufrj.br

Abstract. The main goal of this paper is to present new robust and scalable preconditioned conjugate gradient algorithms for solving Stokes equations with large viscosities jumps across subregion interfaces and discretized on non-structured meshes. The proposed algorithms do not require the construction of a coarse mesh and avoid expensive communications between coarse and fine levels. The algorithms belong to the family of preconditioners based on non-overlapping decomposition of subregions known as balancing domain decomposition methods. The local problems employ two-level element-wise/subdomain-wise direct factorizations to reduce the size and the cost of the local Dirichlet and Neumann Stokes solvers. The Stokes coarse problem is based on subdomain constant pressures and on connected subdomain interface flux functions and rigid body motions. This guaranties scalability and solvability for the local Neumann problems. Estimates on the condition numbers and numerical experiments based on unstructured mesh parallel implementation are also discussed.

1 Introduction

The *core-flow* technique is a technology in research that can turn much more efficient the production/transportation through pipe of heavy oil. The numerical simulators available nowadays are inefficient for solving large scale problems with high jump in viscosity such as the core-flow model. In order to develop an efficient parallel code to solve such model, we develop a preconditioner for the Stokes problem that is robust with respect to high jump in viscosity and are suitable for unstructured meshes.

Balancing Domain Decomposition (BDD) methods are preconditioners based on non-overlapping decomposition of subregions and they have been tested successfully on several challenging large scale applications [4, 7, 6] and its first scalable version was developed by Mandel [6] for the Poisson equation with the

* This work was supported by ANP/PRH-32. The author is candidate to the best student paper award.

introduction of a coarse problem based on the kernel of the Laplace operator. Extensions of the BDD preconditioner for elliptic problems with possibly large jumps on coefficients were treated subsequently in [2, 9, 10]. The extension of the BDD preconditioner for the Stokes equations had its debut only recently by Pavarino and Widlund [7]. 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 [7] and on this paper.

The goal of this paper 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 $\mathbf{P_2}/\mathrm{P}_0$, have comparable computational costs. The paper is organized as follows. The Sections 2 and 3 present the Stokes equations and the variational formulation, respectively, while on Section 4 we introduce the discretizations used in the numerical experiments. The Section 5 is devoted to the BDD preconditioner for the Stokes equations and the coarse spaces. On Section 6 we present some of the implementation issues, and on Section 7 we provide the numerical results. Section 8 closes the paper with the conclusions.

2 The Stokes Model

Let $\Omega \subset \mathbb{R}^2$ be a domain with a polygonal boundary. We consider the Stokes equations:

$$\begin{cases} -2\nabla \cdot (\nu \boldsymbol{\varepsilon}(\boldsymbol{u})) + \nabla p = \boldsymbol{f} & \text{in } \Omega \\ -\nabla \cdot \boldsymbol{u} = g & \text{in } \Omega \\ \boldsymbol{u} = \boldsymbol{u}_d \text{ on } \partial\Omega \end{cases} \tag{1}$$

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 paper, we assume only Dirichlet boundary condition 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 condition is similar and does not bring any extra difficulties; see also Remark 3.

Remark 1. Since we are assuming Dirichlet boundary condition on all $\partial\Omega$, 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., $\int_\Omega p \ \mathrm{dx} = 0$.

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*

$$\begin{cases} a(\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) = F(\boldsymbol{v}) \ \ \forall \boldsymbol{v} \in \mathbf{X}, \\ b(\boldsymbol{u},q) \qquad\qquad = G(q) \ \ \forall q \in M, \end{cases} \tag{2}$$

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 (2) exists and is unique; see [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 introduce 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 element of reference $\widehat{K}$, 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 [3], i.e., there exists a constant β (independent of h) such that

$$\sup_{\substack{\boldsymbol{v}\in\mathbf{X}_h \\ \boldsymbol{v}\neq 0}} \frac{(\nabla\cdot\boldsymbol{v},q)}{\|\boldsymbol{v}\|_{\mathbf{H^1}}} \geq \beta\|q\|_0 \quad \forall q \in M_h, \tag{3}$$

and so the discrete variational formulation of the Stokes problem (1) given by:

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

$$\begin{cases} a(\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) = F(\boldsymbol{v}) \ \ \forall \boldsymbol{v} \in \boldsymbol{X}_h, \\ b(\boldsymbol{u},q) \qquad\qquad = G(q) \ \ \forall q \in M_h, \end{cases} \tag{4}$$

has a unique solution (see [3]). In matricial form, the discrete linear system (4) is of the form

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}. \tag{5}$$

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 $\Gamma_h^{(i)}$.

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: $\boldsymbol{u}_I$ (the interior velocities), p_I (pressures with zero average in each subdomain Ω_i), $\boldsymbol{u}_\Gamma$ (interface velocities) and p_0 (constant pressure in each Ω_i and with zero average in Ω). Using this reordering, the matrix of the discrete system (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:

$$S \begin{pmatrix} \boldsymbol{u}_\Gamma \\ p_0 \end{pmatrix} = \begin{pmatrix} \widetilde{\boldsymbol{f}}_\Gamma \\ \widetilde{g}_0 \end{pmatrix}, \tag{6}$$

where

$$S = K_{\Gamma\Gamma} - K_{\Gamma I} K_{II}^{-1} K_{I\Gamma} = \begin{pmatrix} S_\Gamma & B_0^T \\ B_0 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} \widetilde{\boldsymbol{f}}_\Gamma \\ \widetilde{g}_0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_\Gamma \\ g_0 \end{pmatrix} - K_{\Gamma I} K_{II}^{-1} \begin{pmatrix} \boldsymbol{f}_I \\ g_I \end{pmatrix}.$$

Remark 2. 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 (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 that K_{II} is the block-diagonal matrix $K_{II} = \text{diag}\{K_{II}^{(1)}, \cdots, K_{II}^{(N)}\}$. 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. The velocity component of ${K_{II}^{(i)}}^{-1}$, denoted by $\mathcal{SH}^{(i)}$, is known as the local discrete Stokes harmonic extension operator with velocity Dirichlet boundary condition prescribed on $\Gamma_h^{(i)}$.

Our goal is to solve the linear system (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.

5.2 BDD Preconditioning

Let us decompose the space $\mathbf{X}_h \times M_h = \left(\oplus_{i=1}^N \mathbf{X}_{i,h} \times M_{i,h}\right) \oplus (\mathbf{V}_{\Gamma,h} \times M_0)$ where $\mathbf{X}_{i,h} = \mathbf{X}_h \cap H_0^1(\Omega_i)$, $M_{i,h} = M_h \cap L_0^2(\Omega_i)$, $\mathbf{V}_{\Gamma,h} = \{\boldsymbol{v} \in \mathbf{X}_h;\ \boldsymbol{v}|_{\Omega_i} = \mathcal{SH}^{(i)}(\boldsymbol{v}|_{\partial\Omega_i}),\ i = 1, \ldots, N\}$, and $M_0 = \{q \in M_h;\ q|_{\Omega_i} = \text{const.},\ i = 1, \ldots, N\}$. We observe that the function $\boldsymbol{v} \in \mathbf{V}_{\Gamma,h}$ is uniquely defined by its value on the interface Γ_h.

We now construct a parallel preconditioner M^{-1} for S in order to make the linear system scalable and well conditioned.

An initial try would be to use an additive Schwarz like preconditioner of the form

$$M^{-1} = \sum_{i=1}^{N} R_i^T D_i^T {S^{(i)}}^{-1} D_i R_i, \tag{7}$$

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 is a diagonal matrix defining a partition of unity on Γ_h, i.e., $\sum_{i=1}^N R_i^T D_i R_i = I$ 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 $D_i = \text{diag}\{\delta_i^{-1}\}$. 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

$$\delta_i = \frac{\sum_{j \in \mathcal{N}_x} \nu_j^\gamma(x)}{\nu_i^\gamma(x)}, \tag{8}$$

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 [9, 10]).

Remark 3. The local problems ${S^{(i)}}^{-1}$ in (7) use natural boundary conditions $\nu_i \nabla \boldsymbol{u} \cdot \boldsymbol{n} - pn = r \quad$ on $\Gamma_h^{(i)}$. In this case the pressure is uniquely determined and therefore the pressure space are now taken on $L^2(\Omega)$.

The preconditioner (7) is not as good as it appears to be. 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

$$S^{(i)} \begin{pmatrix} \boldsymbol{u}_\Gamma^{(i)} \\ p_0^{(i)} \end{pmatrix} = \begin{pmatrix} \widetilde{\boldsymbol{f}}_\Gamma^{(i)} \\ \widetilde{g}_0^{(i)} \end{pmatrix} \tag{9}$$

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 (9) 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 $\mathbf{V}_0 \subset \mathbf{V}_{\Gamma,h}$ to enforce that when solving the linear system (9) the right hand side (RHS) is on the image of $S^{(i)}$, and since $S^{(i)}$ is symmetric, this is equivalent to have 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 5.4.

5.3 Preconditioning in Matricial Form

Let $L_0 : \mathbf{V}_0 \to \Gamma_h$ 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

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

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 (9) are compatible, i.e., for any $\boldsymbol{w} \in \mathbf{V}_{\Gamma,h}$

$$\left\langle \begin{pmatrix} D_i R_i & 0 \\ 0 & * \end{pmatrix} S(I - P_0)\boldsymbol{w}, \boldsymbol{v}_i \right\rangle_{\Gamma_i} = 0 \quad \forall \boldsymbol{v}_i \in \mathrm{Ker}(S^{(i)}). \tag{10}$$

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

$$\sup_{\substack{\boldsymbol{v}_\Gamma \in \mathbf{V}_{\Gamma,h} \\ \boldsymbol{v}_\Gamma \neq 0}} \frac{(\nabla . \mathcal{SH}(\boldsymbol{v}_\Gamma), q_0)^2}{a(\mathcal{SH}\boldsymbol{v}_\Gamma, \mathcal{SH}\boldsymbol{v}_\Gamma)} \geq \beta_0 \|q_0\|^2_{L^2} \ \ \forall q_0 \in M_0. \tag{11}$$

When is the case, as in [7], we can show that the bound for the condition of the preconditioned operator in S-norm is

$$\mathrm{cond}_{S^{1/2}}(T) \leq C(1 + \frac{1}{\beta_0})\frac{1}{\beta^2}(1 + \log(\frac{H}{h}))^2 \tag{12}$$

where β is the inf-sup constant of the original problem (3).

5.4 The Coarse Space

The coarse space $\mathbf{V}_0$ plays an important rule 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

$$\mathbf{V}_0^{(0)} = \begin{array}{l} \text{Rigid Body Motion of each subdomain } \Omega_i \text{ scaled by } \mathrm{diag}\{D_i\} \text{ on } \Gamma_{h,i} \\ \text{and zero on the remaining nodes on } \Gamma_h. \end{array}$$

So, 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 (11). This indicates that the coarse space should be enriched. Since our objective is unstructured 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 unity 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 unity 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. 1):

- for defining $\mathbf{V}_0^{(1)}$ let $w_k^{(1)} \equiv 1$
- for defining $\mathbf{V}_0^{(2)}$ let $w_k^{(2)}(v_j) = 0$ for j even and 1 for j odd
- for defining $\mathbf{V}_0^{(3)}$ let $w_k^{(3)}(v_j) = \min\{d^1_{(j)},\ d^2_{(j)}\}/\text{max\_dist}$
- for defining $\mathbf{V}_0^{(4)}$ let $w_k^{(4)}(v_j) = d^1_{(j)} d^2_{(j)}/(\text{max\_dist})^2$, 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 $\text{max\_dist} = \max_j\{d^1_{(j)},\ d^2_{(j)}\}$.

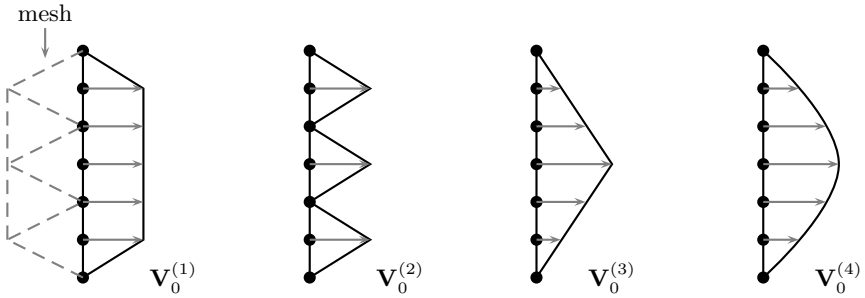

Fig. 1. Sketch of the edge enrichment functions

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.

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 [1] for unstructured meshes. The unstructured meshes are generated using the 2D mesh generator EMC2 from INRIA [8]. The partitioning of the mesh is by elements and it is performed using the ParMETIS library [5].

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 on each $p_I^{(i)}$ in Ω_i and another Lagrange multiplier μ to enforce zero average of p_0 in Ω.

For applying the BDD preconditioner it remains to deal with another issue when solving (9): 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.

6.2 A Higher Order Method

Having implemented the $\mathbf{P_2}/\mathrm{P}_0$ discretization in PETSc we reuse all the *index sets* and *local to global mappings* defined for the $\mathbf{P_2}/\mathrm{P}_0$ to implement the $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$. We add the bubble velocities and the linear average zero pressures on each element $K \in \mathcal{T}_h$, and then, through a static condensation at the element level, we eliminate the bubble functions and the two average zero pressures, resulting in a sort of stabilized $\mathbf{P_2}/\mathrm{P}_0$ finite elements. After solving the linear system we can recover the P_1 discontinuous pressure solution at element level.

7 Numerical Results

A parallel software was developed in C using the PETSc library [1]. 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 7.1 and 7.2 a structured mesh in the domain $[0,1]\times[0,1]$ partitioned into $\sqrt{N}\times\sqrt{N}$ square subdomains. In Subsection 7.3 we consider an unstructured mesh example to study the parallel performance.

For the numerical experiments in Subsections 7.1 and 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-null. Since the preconditioned operator T in (12) is symmetric positive definite with respect to S (see [7]), 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 with two Opteron processors each, where each node has 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.

7.1 Constant Viscosity Tests

In this section all the numerical experiments are performed with a constant viscosity $\nu = 1$ and using the discretization $(\mathbf{P_2} + \mathbf{Bubble})/\mathrm{P}_1$. On Table 1 we fix the mesh of the subdomains to 32×32 and increase the number of subdomains. On Table 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 1 that the coarse spaces $\mathbf{V}_0^{(0)}$ and $\mathbf{V}_0^{(2)}$ do not satisfy the uniform inf-sup stability (11), while the coarse spaces $\mathbf{V}_0^{(3)}$ and $\mathbf{V}_0^{(4)}$ provide scalable algorithms. From Table 2, we see that the iteration counts of all the preconditioners depend very weakly on the size of the local problems. This result is expected due to (12).

Table 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.

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 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.

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)

For the subsequent numerical experiments we consider only the space $\mathbf{V}_0^{(4)}$ since it shows to be the most effective coarse space tested.

On Table 3 we compare the discrete errors of the $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ and the $\mathbf{P_2}/\mathrm{P}_0$ (in parenthesis). 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, 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, 2 for the velocity in the L^2, H^1, *div* norms, and 2 for the pressure in the L^2 norm, respectively.

In Table 4 we compare the discretizations $(\mathbf{P_2} + \mathbf{Bubbles})/\mathrm{P}_1$ and $\mathbf{P_2}/\mathrm{P}_0$ with respect to iteration counts, conditioning, execution and assembling times (given in seconds), and to preconditioning. The table 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 relatively the same for both discretizations. The high CPU time in the case of the local mesh 64×64 will be discussed in Subsection 7.3.

Table 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.

Local mesh	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_0$	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_1$	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{\mathrm{div}}$	$\|p - p_h\|_0$
4×4	3.64e-5 (5.73e-4)	3.88e-3 (3.63e-2)	2.82e-3 (3.31e-2)	1.39e-2 (7.42e-2)
8×8	3.71e-6 (1.47e-4)	9.13e-4 (1.84e-2)	6.93e-3 (1.69e-2)	3.81e-3 (3.72e-2)
16×16	4.13e-7 (3.73e-5)	2.18e-4 (9.26e-3)	1.71e-4 (8.52e-3)	9.78e-4 (1.86e-2)
32×32	4.97e-8 (9.40e-6)	5.39e-5 (4.64e-3)	4.27e-5 (4.27e-3)	2.46e-4 (9.31e-3)
64×64	6.60e-9 (2.36e-6)	1.34e-5 (2.33e-3)	1.07e-5 (2.14e-3)	4.65e-5 (4.65e-3)

Table 4. 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.

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)

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 $\nu_1 = 1$. On Table 5 we provide the number of iterations and the maximum eigenvalue (in parenthesis), for different values of the exponent γ; see (8). The best result is obtained when $\gamma = 1$, although for $\gamma > 1$ the condition numbers present similar behavior. In addition, as predicted in [9, 10], we confirm the strong deterioration on the performance of the algorithms when γ is less than $1/2$ and ν_2 is large.

Table 5. Fixing the Number of subdomains to 4x4

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)

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 (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 6 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 $S_\mathrm{N} = T_1/T_\mathrm{N}$. 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 7 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 1.

Fig. 2. Domain for parallel performance test and sketch of an unstructured mesh.

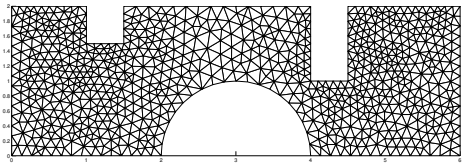

Table 6. This table shows the iteration counts (It.), total execution time (T_{tot}), the speedup factors, the efficiency, and the CPU times to solve iteratively the linear system (T_{S}), to compute the LU factorizations of the local problems (T_{F}) and to compute the coarse matrix (this includes the 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	Efficiency	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

8 Conclusions

We propose four coarse spaces suitable for BDD preconditioning on unstructured meshes. It is verified 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 show that the discretization $(\mathbf{P_2} + \mathbf{Bubble})/\mathrm{P}_1$ is much more accurate than the $\mathbf{P_2}/\mathrm{P}_0$, with no significant extra computational cost. We have shown that the choice $\gamma \geq 1$ in the definition of the diagonal scaling (8) is a robust choice for highly discontinuous viscosities.

Table 7. The local mesh is fixed to 3222 elements. The legends are the same as in the Table 6.

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

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.

Acknowledgements: D. Conceição acknowledges to all the PETSc support for valuable suggestions, and to IMPA's cluster support S. Pilotto and D. Albuquerque.

References

1. Balay, S., Buschelman, K., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F., Zhang, H.: PETSc Web page : http://www.mcs.anl.gov/petsc (2001).
2. Dryja, M., Widlund, O.: Schwarz Methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. Comm. Pure Appl. Math. **48** (1995), no.2, 121–155.
3. Girault, V., Raviart, P.-A.: Finite element methods for Navier-Stokes equations: Theory and algorithms. Springer Series in Computational Mathematics, 5. Springer, Berlin, (1986).
4. Goldfeld, P., Pavarino, L.F., Widlund, O.: Balancing Neumann-Neumann preconditioners for mixed approximations of heterogeneous problems in linear elasticity. Numer. Math. **95** (2003), no. 2, 283–324.
5. Karypis, G., Schloegel, K., Kumar, V.: ParMETIS – Parallel Graph Partitioning and Sparse Matrix Ordering Library. Version 3.1. Web page: http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview
6. Mandel, J,: Balancing domain decomposition, Comm. Appl. Numer. Methods **9** (1993) 233–241.
7. Pavarino, L.F., Widlund, O.: Balancing Neumann-Neumann methods for incompressible Stokes equations. Comm. Pure Appl. Math. **55** (2002), no.3, 302–335.
8. Saltel, E., Hecht, F.: EMC2 Wysiwyg 2D finite elements mesh generator. INRIA. EMC2 web page: http://www-rocq1.inria.fr/gamma/cdrom/www/emc2/eng.htm
9. Sarkis, M.: Two-level Schwarz methods for nonconforming finite elements and discontinuous coefficients, Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods, N.D. Melson, T.A. Manteuffel and S.F. McCormick, eds., Vol. 2, no. 3224, 543–566, NASA, Hampton VA, 1993.
10. Sarkis, M.: Nonstandard coarse spaces and Schwarz methods for elliptic problems with discontinuous coefficients using non-conforming element, Num. Math., **77** (1997) 383–406.