Saddle point preconditioners for linearized Navier–Stokes equations discretized by a finite volume method

Sarah Delcourte *

Institut Camille Jordan, Université Claude Bernard - Lyon 1, France

Delphine Jennequin

Dassault-systèmes, Vélizy, France

Abstract

We are interested in the numerical resolution of the Navier–Stokes problem discretized by a finite–volume method, named DDFV (Discrete Duality Finite Volume method), whose particularity is to work with unstructured and non-conforming meshes. The initial nonlinear system may be transformed into a sequence of Oseen–type problems in rotational form which take the form of saddle–point problems. Some efficient preconditioners have been studied when this kind of Oseen-type problem is discretized by a finite element or a finite difference method but they have not been extended to matrices obtained from finite volume methods. In the present work, we will focus on the adaptation of the most efficient preconditioning techniques to our finite–volume scheme.

Key words: Finite volume methods, saddle-point problem preconditioners, Oseen problem
PACS: 65F10, 76M12, 76D07

* Adresse complete

Email addresses: delcourte@math.univ-lyon1.fr (Sarah Delcourte), delphine.jennequin@3ds.com (Delphine Jennequin).

1 Introduction

Let Ω be a bounded domain of $\mathbb{R}^2$ with Lipschitz boundary Γ. We consider the Navier–Stokes equations with homogeneous Dirichlet boundary conditions:

$$\begin{cases} -\nu \boldsymbol{\Delta} \mathbf{u} + \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} + \boldsymbol{\nabla} p = \mathbf{f} \text{ in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 \text{ in } \Omega, \\ \mathbf{u} = 0 \text{ on } \Gamma, \\ \int_\Omega p(\mathbf{x}) \, d\mathbf{x} = 0, \end{cases} \tag{1}$$

where the viscosity ν and the function $f \in L^2(\Omega)$ are given. The unknowns $\mathbf{u}$ and p represent the velocity field and the pressure of the fluid. Bold fonts are used to represent vectorial variables and operators.

Note that the right-hand side satisfies the following condition:

$$\int_\Omega \nabla \cdot \mathbf{u}(\mathbf{x}) \, d\mathbf{x} = \int_\Gamma \mathbf{u}(\xi) \cdot \mathbf{n}(\xi) \, d\xi = 0. \tag{2}$$

The rotational formulations are well suited for discretizations on staggered unstructured grids. For continuous operators, $-\boldsymbol{\Delta} \mathbf{u}$ can be rewritten by

$$-\boldsymbol{\Delta} \mathbf{u} = \boldsymbol{\nabla} \times \nabla \times \mathbf{u} - \boldsymbol{\nabla} \nabla \cdot \mathbf{u} \, .$$

Note that, in 2D, we should make a distinction between the scalar curl $\nabla \times \mathbf{u} := \dfrac{\partial \mathbf{u}_y}{\partial x} - \dfrac{\partial \mathbf{u}_x}{\partial y}$ and the vectorial curl $\boldsymbol{\nabla} \times \phi = \left(\dfrac{\partial \phi}{\partial y}, \dfrac{-\partial \phi}{\partial x} \right)^T$. On the other hand, we apply the rotational formulation of $\mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u}$:

$$\mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} = (\nabla \times \mathbf{u}) \, \mathbf{u} \times \mathbf{e}_z + \boldsymbol{\nabla} \left(\frac{\mathbf{u}^2}{2} \right) , \tag{3}$$

for which we associate the Bernoulli pressure:

$$\pi = p + \frac{\mathbf{u}^2}{2} \, . \tag{4}$$

At last, in order to ensure the uniqueness of π, we enforce $\int_\Omega \pi(\mathbf{x}) \, d\mathbf{x} = 0$. Therefore, the problem

(1) can be changed into: given $\mathbf{f}$, find $(\mathbf{u}, \pi)$ such that

$$\begin{cases} \nu \ [\boldsymbol{\nabla} \times \nabla \times \mathbf{u} - \boldsymbol{\nabla} \nabla \cdot \mathbf{u}] + (\nabla \times \mathbf{u}) \ \mathbf{u} \times \mathbf{e}_z + \boldsymbol{\nabla} \pi = \mathbf{f}, & \text{in } \Omega \\ \nabla \cdot \mathbf{u} = 0, & \text{in } \Omega, \\ \mathbf{u} = 0, & \text{on } \Gamma, \\ \int_\Omega \pi(\mathbf{x}) \ d\mathbf{x} = 0. \end{cases} \tag{5}$$

When the Navier-Stokes equations are solved by a fixed–point type method (see [22]), we must solve a linear system to each non–linear iteration. Hence, at each nonlinear step, we have to solve the Oseen-type problem: for given $\mathbf{f}$ and $\mathbf{u}_G$,

$$\begin{cases} \nu \, [\boldsymbol{\nabla} \times \nabla \times \mathbf{u} - \boldsymbol{\nabla} \nabla \cdot \mathbf{u}] + (\nabla \times \mathbf{u}_G) \ \mathbf{u} \times e_z + \boldsymbol{\nabla} \pi = \mathbf{f} \text{ in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 \text{ in } \Omega, \\ \mathbf{u} = 0 \text{ on } \Gamma, \\ \int_\Omega \pi(\mathbf{x}) \ d\mathbf{x} = 0, \end{cases} \tag{6}$$

where $\mathbf{u}_G$ is known from the previous non-linear step. We will discretize the linear problem (6) by the finite volume scheme in discrete duality as described in [7,4], which gives a discretization well suited for arbitrary meshes. The meshes are built as follows: at first, we start from a polygonal primal mesh and next, we define the dual mesh as composed of polygons whose vertices are the circumcenters of the adjacent primal triangles and the midpoints of the adjacent primal edges. At last, we consider a third mesh composed of quadrilateral cells (called diamond cells) whose vertices are the extremities of primal and associated dual edges. Thanks to these meshes, we can compute the discrete gradient and the discrete vectorial curl of a function by their values over the diamond cells; the discrete divergence and the discrete scalar curl are defined by their values over the primal and the dual cells. The discretization by finite volume scheme in discrete duality leads to a system–matrix which takes the form of the following saddle–point problem

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \pi \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}, \tag{7}$$

where A is a convection–diffusion–type matrix, B and B^T are the discrete divergence and gradient matrices. The discrete unknown $\mathbf{u}$ corresponds to the approximation of the velocity on the diamond cells and the discrete pressure π is the approximation of the Bernoulli pressure both on the primal and dual meshes.

The numerical solutions of saddle–point problems have become a center of interest in recent years: they appear in many applications of scientific computing and engineering (see the survey [16]). As

it is well known, saddle–point systems can be solved by an Uzawa method or by a block–triangular preconditioner applied on the complete system matrix inside a Krylov method [14,20]. However, the real difficulty of the saddle–point problems resolution is to find an efficient preconditioner for the Schur complement $S := -BA^{-1}B^T$.

Cahouet and Chabard [2] first introduced a way to precondition the Schur complement of the Stokes saddle–point problem discretized by LBB stable finite elements; they used the formal commutation of the operators

$$BA^{-1}B^T = BB^T A_p^{-1},$$

where A_p is the Laplacian matrix on the pressure space, associated to Neumann boundary conditions. Since BB^T corresponds formally to a discretization of $-\nabla \cdot \nabla = -\Delta$, they approached the Schur complement by the scaled mass matrix

$$-\nu^{-1}Q. \tag{8}$$

This preconditioner was also used for Oseen–type problems. It has been proved in [12] that the number of iterations of the preconditioned problem is independent of the mesh size when the problem is discretized by stable finite elements; this result remains true for some stabilized discretizations [21,3]. In [18], Olshanskii studied a preconditioner for the Navier–Stokes equation in rotational form. He proved that, when the problem is discretized by LBB–stable finite elements, applying the preconditioner (8) leads also to an iteration count independent of the mesh size. When this preconditioner is used for Oseen or Navier–Stokes problems, the number of iterations increases linearly with $1/\nu$ and this preconditioner is only efficient for a very moderate Reynolds numbers [15]. To take into account the non symmetry of the convection–diffusion matrix A, Elman [10] introduced a preconditioner for the Navier–Stokes problem discretized by MAC finite difference method. He proposed to use

$$-(BB^T)(BAB^T)^{-1}(BB^T). \tag{9}$$

The number of iterations of the preconditioned saddle–point problem is mildly dependent of ν and increases in proportion to $h^{-1/2}$ where h represents the mesh size. Moreover, for constant wind in Oseen-type problem, Elman showed that the iteration count is independent of the viscosity. However, $-BB^T$ is exactly the Laplacian matrix on a MAC scheme, but this property is not necessarily true for finite element or finite volume methods. In [13,20,12], the authors propose to precondition the Schur complement issue from the Navier–Stokes problems discretized by LBB-stable finite elements by

$$-HA_p^{-1}Q, \tag{10}$$

where H is the Laplacian matrix, A_p the convection–diffusion matrix on the pressure space and Q the mass–matrix. With this preconditioner, the iteration count is independent of the mesh size and depends of order $\nu^{-1/2}$ on the viscosity. The main drawback of this preconditioner consists in the computation of the convection–diffusion matrix on the pressure space : for example, this is not possible with the finite–volume method in discrete duality because of the unknowns locations. In [9], Elman et al. extended the notion of formal commutators developed in [10]. This new preconditioner takes the form of

$$-(BM_2^{-1}B^T)(BM_2^{-1}AM_1^{-1}B^T)^{-1}(BM_1^{-1}B^T), \tag{11}$$

where M_1 and M_2 are weighted matrices to be defined. This method can be viewed as an algebraic extension of preconditioners (9) and (10) : indeed, these preconditioners can be found with appropriate choice of M_1 and M_2. This method is particularly interesting since it can be used as a black–box preconditioner. More recently, Olshanskii et al. [19] proposed to use

$$-Q(BL^{-1}AL^{-1}B^T)^{-1}Q, \tag{12}$$

where L corresponds to the diffusion part of the matrix A. They proved that their preconditioner leads to iterative count independent on the mesh size and dependent like ν^{-1} on the viscosity.

This paper is organized as follows: in section 2, we will define the primal, dual and diamond meshes with associated discrete gradient, divergence and curl operators. Then, section 3 will be devoted to the discretization of the Oseen problem. After, we will describe some iterative solvers in section 4 and at last, some numerical results will be presented in section 5.

2 Definitions and notations

2.1 Construction of the primal mesh

We consider a first partition of Ω (named primal mesh) composed of elements T_i, with $i \in [1, I]$, supposed to be convex polygons. To each element T_i of the mesh is associated a node G_i located at the barycentre of T_i. We assume in what follows that each frontier T_i has only one edge belonging to the boundary. The area of T_i is denoted by $|T_i|$. We shall denote by J the total number of sides of this mesh and by J^Γ the number of these edges which are located on the boundary Γ and we associate with each of these boundary edges its midpoint, also denoted by G_i with $i \in [I+1, I+J^\Gamma]$.

2.2 Construction of the dual mesh

Further, we denote by S_k, with $k \in [1, K]$, the nodes of the polygons of the primal mesh. To each of these points, we associate a polygon denoted by P_k, obtained by joining the points G_i associated to the elements of the primal mesh (and possibly to the boundary sides) of which S_k is a vertex, to the midpoints of the edges of which S_k is an extremity. The area of P_k is denoted by $|P_k|$. The P_ks constitute a second partition of Ω, referenced as dual mesh. Figure 1 displays an example of a primal mesh and its associated dual mesh.
Moreover, we suppose that the set $[1, K]$ is ordered so that when S_k is not on Γ, then $k \in [1, K - J^\Gamma]$, and when S_k is on Γ, then $k \in [K - J^\Gamma + 1, K]$.

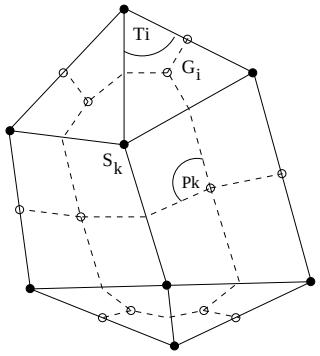


Figure 1. An example of a primal mesh and its associated dual mesh.

2.3 Construction of the diamond mesh

With each side of the primal mesh, denoted by A_j (whose length is $|A_j|$), with $j \in [1, J]$, we associate a quadrilateral named "diamond cell" and denoted by D_j. When A_j is not on the boundary, this cell is obtained by joining the points $S_{k_1(j)}$ and $S_{k_2(j)}$, which are the two nodes of A_j, with the points $G_{i_1(j)}$ and $G_{i_2(j)}$ associated to the elements of the primal mesh which share this side. When A_j is on the boundary Γ, the cell D_j is obtained by joining the two nodes of A_j with the point $G_{i_1(j)}$ associated to the only element of the primal mesh of which A_j is a side and to the point $G_{i_2(j)}$ associated to A_j (*i.e.* by convention $i_2(j)$ is element of $[I+1, I+J^\Gamma]$ when A_j is located on Γ). The cells D_j constitute a third partition of Ω, which we name "diamond-mesh". The area of the cell D_j is denoted by $|D_j|$. Such cells are displayed on figure 2.
Moreover, we suppose that the set $[1, J]$ is ordered so that when A_j is not on Γ, then $j \in [1, J - J^\Gamma]$, and when A_j is on Γ, then $j \in [J - J^\Gamma + 1, J]$.

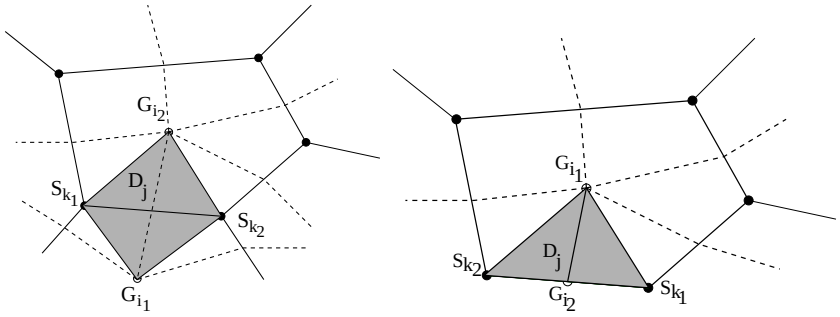


Figure 2. Examples of diamond cells.

2.4 Definitions of geometrical elements

The unit vector normal to A_j is denoted by $\mathbf{n}_j$ and is oriented so that $\mathbf{G_{i_1(j)}G_{i_2(j)}} \cdot \mathbf{n}_j \geq 0$. We further denote by A'_j the segment $[G_{i_1(j)}G_{i_2(j)}]$ (whose length is $|A'_j|$) and by $\mathbf{n}'_j$ the unit vector

normal to A'_j oriented so that $\mathbf{S_{k_1(j)}S_{k_2(j)}} \cdot \mathbf{n}'_j \geq 0$. We define for each $i \in [1, I]$ the set $V(i)$ of integers

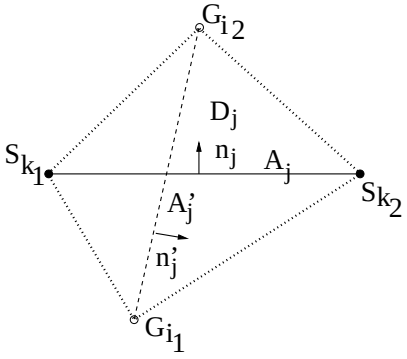


Figure 3. Notations for the diamond cell.

$j \in [1, J]$ such that A_j is a side of T_i and for each $k \in [1, K]$ the set $E(k)$ of integers $j \in [1, J]$ such that S_k is a node of A_j.
We define for each $j \in [1, J]$ and each k such that $j \in E(k)$ (resp. each i such that $j \in V(i)$) the real-valued number s'_{jk} (resp. s_{ji}) whose value is $+1$ or -1 whether $\mathbf{n}'_j$ (resp. $\mathbf{n}_j$) points outwards or inwards P_k (resp. T_i). We define $\mathbf{n}'_{jk} := s'_{jk}\mathbf{n}'_j$ (resp. $\mathbf{n}_{ji} := s_{ji}\mathbf{n}_j$) and remark that $\mathbf{n}'_{jk}$ (resp. $\mathbf{n}_{ji}$) always points outwards P_k (resp. T_i).

2.5 Definition of the operators

We may approach the gradient operator $\boldsymbol{\nabla}\bullet = \left(\frac{\partial \bullet}{\partial \mathbf{x}}, \frac{\partial \bullet}{\partial \mathbf{y}}\right)$ by a discrete gradient operator on the diamond cells D_j (see [7]).

Definition 1 *Given any $\phi = (\phi_i^T, \phi_k^P) \in \mathbb{R}^{I+J^\Gamma} \times \mathbb{R}^K$, the discrete gradient $\boldsymbol{\nabla}_h^D$ is defined by its values over the diamond cells D_j:*

$$(\boldsymbol{\nabla}_h^D \phi)_j := \frac{1}{2\,|D_j|}\left\{\left[\phi_{k_2}^P - \phi_{k_1}^P\right]|A'_j|\mathbf{n}'_j + \left[\phi_{i_2}^T - \phi_{i_1}^T\right]|A_j|\mathbf{n}_j\right\}, \tag{13}$$

if we set $\phi_k^P := \phi(S_k)$ and $\phi_i^T := \phi(G_i)$, for any (i, k).

The operator $\boldsymbol{\nabla}_h^D$ thus acts from $\mathbb{R}^{I+J^\Gamma} \times \mathbb{R}^K$ into $\left(\mathbb{R}^J\right)^2$.

In the very same way, we may approach the vector curl operator $\boldsymbol{\nabla} \times \bullet$ by a discrete vector curl operator:

Definition 2 *Given any $\phi = (\phi_i^T, \phi_k^P) \in \mathbb{R}^{I+J^\Gamma} \times \mathbb{R}^K$, the discrete vector curl operator $\boldsymbol{\nabla}_h^D \times$ is*

defined by its values over the diamond cells D_j:

$$(\boldsymbol{\nabla}_h^D \times \phi)_j := -\frac{1}{2\,|D_j|}\left\{\left[\phi_{k_2}^P - \phi_{k_1}^P\right]|A_j'|\boldsymbol{\tau}_j' + \left[\phi_{i_2}^T - \phi_{i_1}^T\right]|A_j|\boldsymbol{\tau}_j\right\}, \tag{14}$$

where the unit vectors $\boldsymbol{\tau}_j$ and $\boldsymbol{\tau}_j'$ are such that $(\mathbf{n}_j, \boldsymbol{\tau}_j)$ and $(\mathbf{n}_j', \boldsymbol{\tau}_j')$ are orthogonal positively oriented bases of $\mathbb{R}^2$.

Next, we choose to define the discrete divergence of a vector field $\mathbf{u}$ by its values both on the primal and dual cells of the mesh. Supposing that the vector field $\mathbf{u}$ is given by its discrete values $\mathbf{u}_j$ on the cells D_j, we state *the definition* of the discrete divergence $\nabla_h^T\cdot$ on each T_i and the discrete divergence $\nabla_h^P\cdot$ on each P_k.

Definition 3 *Given any $\mathbf{u} = (\mathbf{u}_j) \in \mathbb{R}^{2J}$, the discrete divergence $\nabla_h^{T,P}\cdot := (\nabla_h^T\cdot, \nabla_h^P\cdot)$ is defined by its values over the primal cells T_i and the dual cells P_k (see Fig. 4)*

$$\begin{aligned}
(\nabla_h^T \cdot \mathbf{u})_i &:= \frac{1}{|T_i|}\sum_{j\in V(i)} |A_j|\mathbf{u}_j\cdot\mathbf{n}_{ji}\,, \\
(\nabla_h^P \cdot \mathbf{u})_k &:= \frac{1}{|P_k|}\left(\sum_{j\in E(k)}\left(|A_{j1}'|\mathbf{n}_{jk1}' + |A_{j2}'|\mathbf{n}_{jk2}'\right)\cdot\mathbf{u}_j + \sum_{j\in E(k)\cap[J-J^\Gamma+1,J]}\frac{1}{2}\,|A_j|\mathbf{u}_j\cdot\mathbf{n}_j\right),
\end{aligned} \tag{15}$$

where we recall that $V(i)$ (resp. $E(k)$) is the set of integers $j \in [1, J]$ such that A_j is a side of T_i (resp. S_k is a node of A_j) and that $\mathbf{n}_{ji}$ (resp. $\mathbf{n}_{jk1}'$ and $\mathbf{n}_{jk2}'$) is the unit vector orthogonal to A_j (resp. A_{j1}' and A_{j2}') pointing outward T_i (resp. P_k).

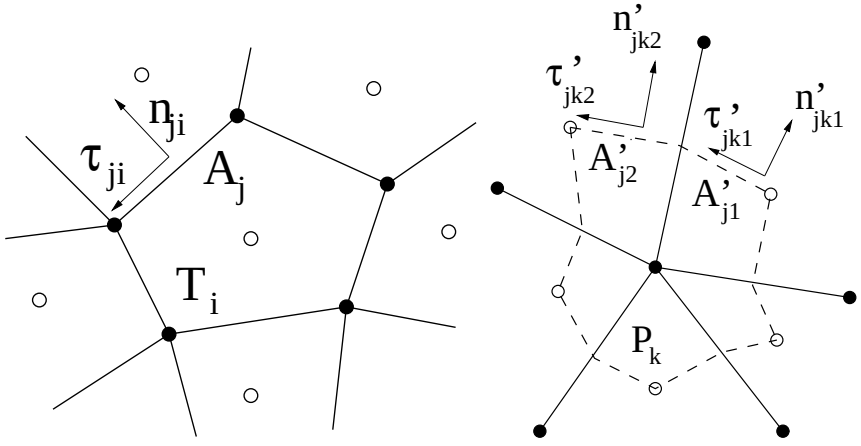


Figure 4. Edges and unit vectors for the discrete divergence and curl

For more details about the discrete operators, refer to [7,4].

For a given vector field $\mathbf{u}$, it is straightforward to check that formulae (15) are the exact mean-values of $\nabla \cdot \mathbf{u}$ over T_i, respectively over an inner P_k, if

$$|A_j| \mathbf{u}_j \cdot \mathbf{n}_{ji} = \int_{A_j} \mathbf{u} \cdot \mathbf{n}_{ji} \, ds \ ,$$

resp. if

$$\left(|A'_{j1}| \mathbf{n}'_{jk1} + |A'_{j2}| \mathbf{n}'_{jk2} \right) \cdot \mathbf{u}_j = \int_{A'_{j1}} \mathbf{u} \cdot \mathbf{n}'_{jk1} \, ds + \int_{A'_{j2}} \mathbf{u} \cdot \mathbf{n}'_{jk2} \, ds.$$

In the very same way, we may approach the scalar curl operator $\nabla \times \bullet = \left(\frac{\partial \bullet_y}{\partial x} - \frac{\partial \bullet_x}{\partial y} \right)$ by a discrete scalar curl operator:

Definition 4 *Given any $\mathbf{u} = (\mathbf{u}_j) \in \mathbb{R}^{2J}$, the discrete scalar curl operator $\nabla_h^{T,P} \times := (\nabla_h^T \times, \nabla_h^P \times)$ is defined by its values over the primal cells T_i and the dual cells P_k:*

$$\begin{aligned} (\nabla_h^T \times \mathbf{u})_i &:= \frac{1}{|T_i|} \sum_{j \in V(i)} |A_j| \mathbf{u}_j \cdot \boldsymbol{\tau}_{ji} \ , \\ (\nabla_h^P \times \mathbf{u})_k &:= \frac{1}{|P_k|} \left(\sum_{j \in E(k)} \left(|A'_{j1}| \boldsymbol{\tau}'_{jk1} + |A'_{j2}| \boldsymbol{\tau}'_{jk2} \right) \cdot \mathbf{u}_j + \sum_{j \in E(k) \cap [J - J^\Gamma + 1, J]} \frac{1}{2} \, |A_j| \mathbf{u}_j \cdot \boldsymbol{\tau}_j \right) . \end{aligned} \tag{16}$$

We can also define some discrete scalar products :

Definition 5 *(The discrete scalar products) Let $(\phi, \psi) \in \left(\mathbb{R}^I \times \mathbb{R}^K \right)^2$ and $(\mathbf{u}, \mathbf{v}) \in \left(\mathbb{R}^{2J} \right)^2$, then we define the following scalar products:*

$$(\mathbf{u}, \mathbf{v})_D := \sum_{j \in [1,J]} |D_j| \, \mathbf{u}_j \cdot \mathbf{v}_j, \tag{17}$$

$$(\phi, \psi)_{T,P} := \frac{1}{2} \left(\sum_{i \in [1,I]} |T_i| \, \phi_i^T \, \psi_i^T + \sum_{k \in [1,K]} |P_k| \, \phi_k^P \, \psi_k^P \right), \tag{18}$$

We also define the trace of $u \in \mathbb{R}^J$ and $\phi \in \mathbb{R}^I \times \mathbb{R}^K$ on the boundary Γ by

$$(u, \phi)_{\Gamma,h} := \sum_{j \in \Gamma} |A_j| \, u_j \times \frac{1}{4} \left(\phi^P_{k_1(j)} + 2\phi^T_{i_2(j)} + \phi^P_{k_2(j)} \right). \tag{19}$$

These scalar products are built such that the discrete gradient, divergence and curl operators satisfy some discrete duality principles, expressed in the following proposition.

Proposition 6 *(The discrete Green formulae) The following discrete analogues of the Green formulae hold:*

$$(\nabla_h^{T,P} \cdot \mathbf{u}, \phi)_{T,P} = -(\mathbf{u}, \nabla_h^D \phi)_D + (\mathbf{u} \cdot \mathbf{n}, \phi)_{\Gamma,h} \,, \tag{20}$$

$$(\nabla_h^{T,P} \times \mathbf{u}, \phi)_{T,P} = (\mathbf{u}, \nabla_h^D \times \phi)_D + (\mathbf{u} \cdot \tau, \phi)_{\Gamma,h} \,, \tag{21}$$

for all $\mathbf{u} \in \left(\mathbb{R}^J\right)^2$ *and all* $\phi = (\phi^T, \phi^P) \in \mathbb{R}^{I+J^\Gamma} \times \mathbb{R}^K$*, where the definitions (17), (18) and (19) have been used.*

3 Discretization of the Oseen equations

3.1 Discrete system

Discretization and linearisation with an iterative process (as fixed–point–like methods (see [22])) leads us to solve the following linear system, called Oseen equations: given $\mathbf{f}$ and $\mathbf{u}_G$, find $(\mathbf{u}, \pi)$ such that

$$\begin{cases} \nu \ [\boldsymbol{\nabla} \times \nabla \times \mathbf{u} - \boldsymbol{\nabla} \nabla \cdot \mathbf{u}] + (\nabla \times \mathbf{u}_G) \ \mathbf{u} \times \mathbf{e}_z + \boldsymbol{\nabla} \pi = \mathbf{f} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = 0 & \text{in } \Gamma, \\ \displaystyle\int_\Omega \pi(\mathbf{x}) \ d\mathbf{x} = 0. \end{cases} \tag{22}$$

More precisely, we will compute the approximation $(\mathbf{u}_j)_{j\in[1,J]}$ of the velocity $\mathbf{u}$ on the diamond cells and the approximation $(\pi_i^T)_{i\in[1,I]}$, $(\pi_k^P)_{k\in[1,K]}$ of the Bernoulli pressure π on the primal and dual cells respectively.

We discretize the first equation of (22) on the interior diamond cells, and the second equation both on the primal and dual cells. Then, the boundary condition $\mathbf{u} = 0$ is discretized on the boundary diamond cells while the condition of vanishing mean pressure is discretized on the primal and dual cells.

We suppose that the locations of the values of $\mathbf{u}_G$ are the same as those of $\mathbf{u}$, that is on the diamond cells. Therefore, we may easily calculate $\nabla \times \mathbf{u}_G$ on the primal and dual cells according to the discrete operator $\nabla_h^{T,P}\times$. However, since the first equation in (22) is discretized on diamond cells, we shall use the following quadrature formula to calculate $\nabla \times \mathbf{u}_G$ over any D_j:

$$(\nabla \times \mathbf{u}_G)_{|_{D_j}} \approx \frac{(\nabla_h^T \times \mathbf{u}_G)_{i_1} + (\nabla_h^T \times \mathbf{u}_G)_{i_2} + (\nabla_h^P \times \mathbf{u}_G)_{k_1} + (\nabla_h^P \times \mathbf{u}_G)_{k_2}}{4} \,. \tag{23}$$

Then, for all diamond cells, we set:

$$-\left[\boldsymbol{\Delta}_h^D \mathbf{u}\right]_j = (\boldsymbol{\nabla}_h^D \times \nabla_h^{T,P} \times \mathbf{u})_j - (\boldsymbol{\nabla}_h^D \nabla_h^{T,P} \cdot \mathbf{u})_j.$$

Now, we can discretize the continuous problem (22) by the following system:

$$-\nu \ \left[\boldsymbol{\Delta}_h^D \mathbf{u}\right]_j + (\nabla \times \mathbf{u}_G)_{|_{D_j}} \mathbf{u}_j \times \mathbf{e}_z + (\boldsymbol{\nabla}_h^D \pi)_j = \mathbf{f}_j^D, \quad \forall D_j \notin \Gamma, \tag{24a}$$

$$(\nabla_h^{T,P} \cdot \mathbf{u})_{i,k} = 0, \quad \forall T_i, \forall P_k, \tag{24b}$$

$$\mathbf{u}_j = \mathbf{f}_j^D, \quad \forall D_j \in \Gamma, \tag{24c}$$

$$\sum_{i \in [1,I]} |T_i| \ \pi_i^T = \sum_{k \in [1,K]} |P_k| \ \pi_k^P = 0, \tag{24d}$$

where we have set $\mathbf{f}_j^D = \dfrac{1}{|D_j|} \displaystyle\int_{D_j} \mathbf{f}(\mathbf{x}) \ d\mathbf{x}$, $\forall j \notin \Gamma$ and $\mathbf{f}_j^D = 0$, $\forall j \in \Gamma$.

Theorem 1 *If each boundary primal cell has only one edge which belongs to the boundary, then the solution* $((\mathbf{u}_j)_{j \in [1,J]}, (\pi_i^T)_{i \in [1,I]}, (\pi_k^P)_{k \in [1,K]})$ *of (24) exists and is unique.*

The proof is given in [6]. Once $(\mathbf{u}_j)_{j \in [1,J]}$ and $(\pi_i^T, \pi_k^P)_{i \in [1,I], k \in [1,K]}$ have been computed, we can deduce the physical pressure from the Bernoulli pressure by integration and projection. For more details, refer to [5].

3.2 Algebraic formulation

In this section, we give the values of the matrices A, B and B^T related to (24a)–(24c) which form the saddle–point problem:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \pi \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ 0 \end{pmatrix}.$$

Note that, using the Green formula (20), we can prove that the operators $-\nabla_h^{T,P} \cdot$ and $\boldsymbol{\nabla}_h^D$ are adjoint when $\mathbf{u}$ satisfies homogeneous Dirichlet boundary conditions.
The $2J \times 2J$ matrix A is defined by $A = \nu L + N$, where

$$(L\bullet)_j = \begin{cases} |D_j| \left[(\boldsymbol{\nabla}_h^D \times \nabla_h^{T,P} \times \bullet)_j - (\boldsymbol{\nabla}_h^D \nabla_h^{T,P} \cdot \bullet)_j \right], \forall j \notin \Gamma, \\ |D_j| \ \bullet_j \qquad\qquad\qquad\qquad\qquad\quad , \forall j \in \Gamma, \end{cases} \tag{25}$$

and

$$(N\bullet)_j = \begin{cases} |D_j| \ (\nabla \times \mathbf{u}_G)_{|_{D_j}} (- \bullet_{jy}, \ \bullet_{jx})^T \ , \forall j \notin \Gamma, \\ 0 \qquad\qquad\qquad\qquad\qquad\quad , \forall j \in \Gamma. \end{cases} \tag{26}$$

The $(I+K)\times 2J$ matrix B is defined by

$$(B\bullet)_l = \begin{cases} \frac{-|T_i|}{2}\ (\nabla_h^T \cdot \bullet)_i\ , \text{ if } i = l \in [1, I], \\ \frac{-|P_k|}{2}\ (\nabla_h^P \cdot \bullet)_k\ , \text{ if } k = l - I \in [1, K]. \end{cases} \tag{27}$$

and the vector $\mathbf{F}$ whose size is $2J$ is given by

$$(\mathbf{F}\bullet)_j = \begin{cases} |D_j|\ \mathbf{F}_j\ , \forall j \notin \Gamma, \\ \quad 0 \qquad , \forall j \in \Gamma. \end{cases} \tag{28}$$

Remark 1 *When $\mathbf{u} \neq 0$ on the boundary and $\nabla \cdot \mathbf{u} = 0$ in Ω (see the numerical tests), we perform a change of variables such that $\mathbf{u}_0 = \mathbf{u} - \widetilde{\mathbf{u}}$ which vanihes on the boundary ($\widetilde{\mathbf{u}}$ is the null vector with the boundary conditions only) and, in this case, $B\mathbf{u}_0 \neq 0$ in general. Therefore, we are led to solve a saddle–point problem with a perturbed right-hand side $(\widetilde{\mathbf{F}}, \widetilde{g})$:*

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_0 \\ \pi \end{pmatrix} = \begin{pmatrix} \widetilde{\mathbf{F}} \\ \widetilde{g} \end{pmatrix}. \tag{29}$$

4 Iterative solvers

Using the discrete inner product definition, it makes sense now to define a finite volume analogous of the pressure mass–matrix and of the velocity mass–matrix defined for finite element methods. We denote by Q the pressure mass matrix defined by

$$\forall 1 \leq l \leq I + K, \quad Q_{l,l} = \begin{cases} |T_i| & \text{if } i = l \in [1, I], \\ |P_k| & \text{if } k = l - I \in [1, K]. \end{cases}$$

The velocity mass–matrix (in 2D) X is defined on the diamond cells by

$$\forall 1 \leq l \leq 2J, \quad X_{l,l} = \begin{cases} |D_j| & \text{if } j = l \in [1, J], \\ |D_j| & \text{if } j = l - J \in [1, J]. \end{cases}$$

We compare three kind of preconditioners $\widetilde{S}^{-1}$ of the Schur complement $S = -BA^{-1}B^T$:

- SIMPLE preconditioner: $\widetilde{S}^{-1} = -(B\widehat{A}^{-1}B^T)^{-1}$ where $\widehat{A}$ is an approximation of A (for our tests, we will take for $\widehat{A}$ the diagonal matrix diag(A) whose diagonal is equal to the diagonal of A).

- The approximate commutators (called BFBt preconditioner by Elman [9]):

$$\widetilde{S}^{-1} = -(BM_2^{-1}B^T)^{-1}(BM_2^{-1}AM_2^{-1}B^T)(BM_2^{-1}B^T)^{-1}.$$

  We consider the following candidates for M_2 : the identity matrix I, the mass matrix on the velocity space X and diag(A).
- Olshanskii's preconditioner [19]:

$$\widetilde{S}^{-1} = -Q^{-1}BL^{-1}AL^{-1}B^TQ^{-1},$$

  where L is the Laplacian matrix (25) with homogeneous boundary conditions.

5 Numerical results

In this section, we consider the computational domain $\Omega = [0,1]^2$. Each boundary edge of Ω is decomposed into 20 segments of the same length and we compute the Delaunay triangulation of the domain Ω thanks to the freeware EMC2 [11]. We consider the Oseen problem (influenced by lid-driven cavity benchmark) in the domain Ω with wind

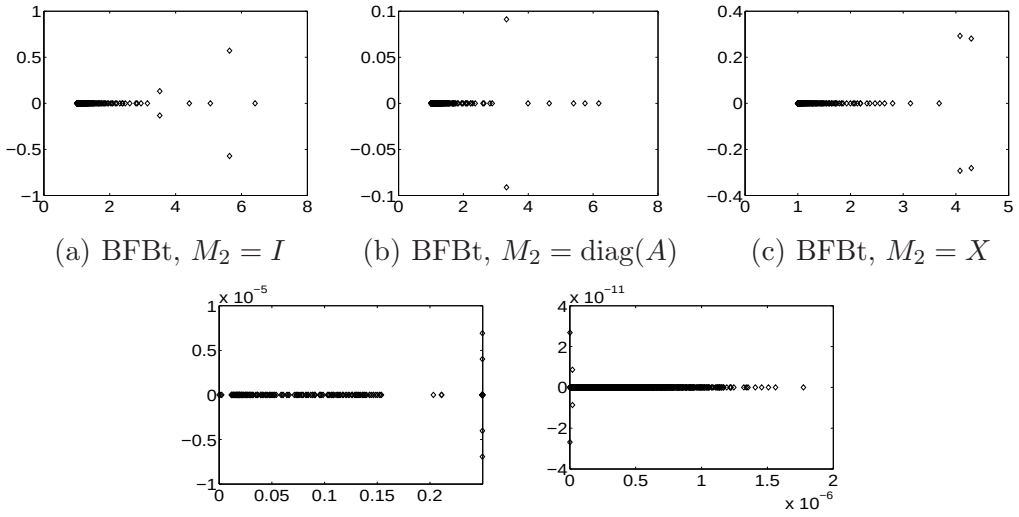
$$w = \begin{pmatrix} 2(2y-1)(1-(2x-1)^2) \\ -2(2x-1)(1-(2y-1)^2) \end{pmatrix}$$

and with boundary conditions $\mathbf{u}(x,1) = (1,0)^T$, $\mathbf{u}(x,y) = 0$ elsewhere. The right-hand side is supposed to be null. With these conditions, the Reynolds number is equal to $\mathcal{R}e = \frac{1}{\nu}$.

5.1 Spectra

Figures 5, 6 7 and 8 show the spectra of the preconditioned Schur complement $\widetilde{S}^{-1}S$ for several values of the viscosity ν and for all preconditioners described above on the unstructured grid 20×20. These results were obtained using the eigs function of matlab. We can observe the clustering effect of the BFBt preconditioner, specially for moderate Reynolds numbers. The spectral condition number is defined by the modulus of the ratio between the eigenvalue of largest real part and the eigenvalue of the smallest real part and it is known that the iteration count of an iterative method applied to the preconditioned problem grows with the spectral condition number. We may observe that, for $\nu \geq 0.01$, the box which contains the eigenvalues is almost constant for the BFBt preconditioner only. We can already predict that this preconditioner gives better results when ν decreases. For $\nu = 0.001$, the smallest eigenvalue tends to zero for all preconditioners: however, these results are not really significant because the mesh size is too large for such a small viscosity. Looking at the spectrum obtained with the Olshankii preconditioner, we can observe that, even for moderate Reynolds number,

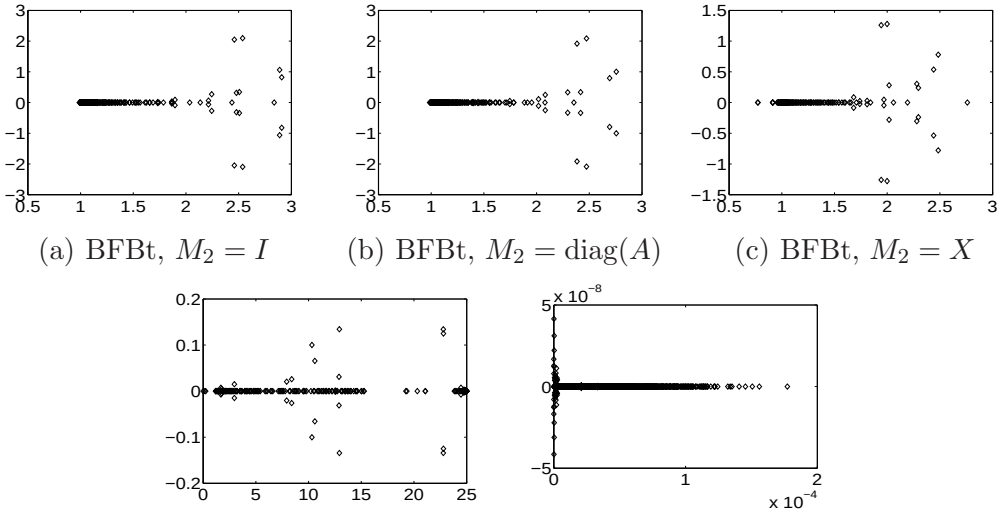
there are some eigenvalues very close to zero : it explains partially the bad performances we had with this preconditioner.



(a) BFBt, $M_2 = I$ (b) BFBt, $M_2 = \mathrm{diag}(A)$ (c) BFBt, $M_2 = X$

(d) Olshanskii (e) SIMPLE

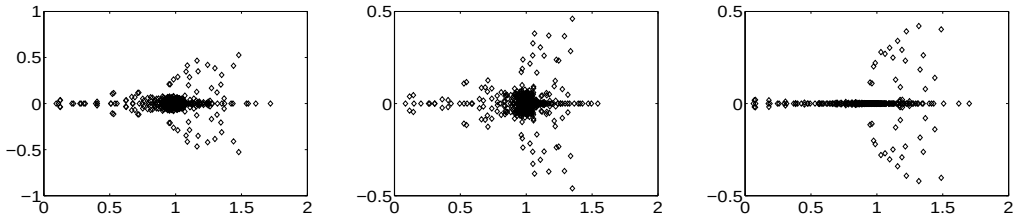
Figure 5. $\nu = 1$. Spectrum of the preconditioned Schur complement



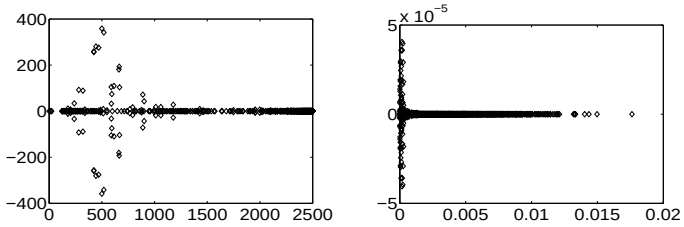
(a) BFBt, $M_2 = I$ (b) BFBt, $M_2 = \mathrm{diag}(A)$ (c) BFBt, $M_2 = X$

(d) Olshanskii (e) SIMPLE

Figure 6. $\nu = 0.1$. Spectrum of the preconditioned Schur complement



(a) BFBt, $M_2 = I$ (b) BFBt, $M_2 = \mathrm{diag}(A)$ (c) BFBt, $M_2 = X$

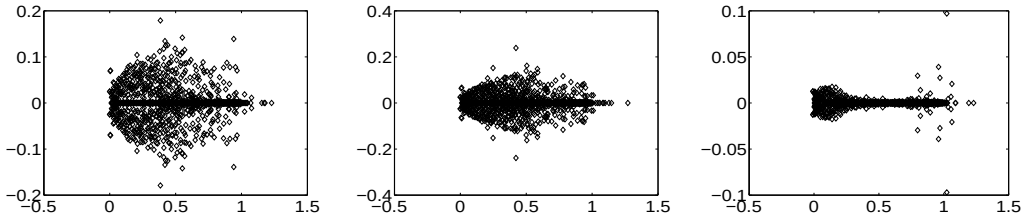


(d) Olshanskii (e) SIMPLE

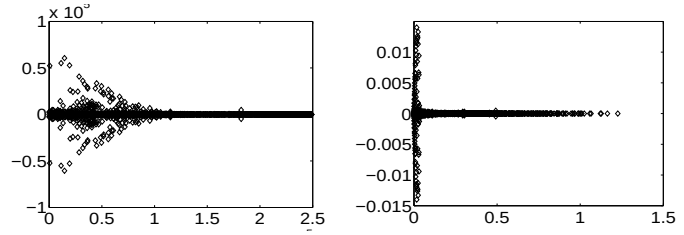
Figure 7. $\nu = 0.01$. Spectrum of the preconditioned Schur complement

5.2 Iteration count

The experiments are done in fortran90 with PETSC [1] and MUMPS libraries [17]. The mesh is generated by EMC2 [11]. The global algorithm is the preconditioned Uzawa-BICGSTAB [23] described in Table 1; it is obtained by applying the classical preconditioned BICGSTAB to the Schur complement system.
For SIMPLE or BFBt preconditioner, we have to solve systems whose matrices take the form



(a) BFBt, $M_2 = I$ (b) BFBt, $M_2 = \mathrm{diag}(A)$ (c) BFBt, $M_2 = X$



(d) Olshanskii (e) SIMPLE

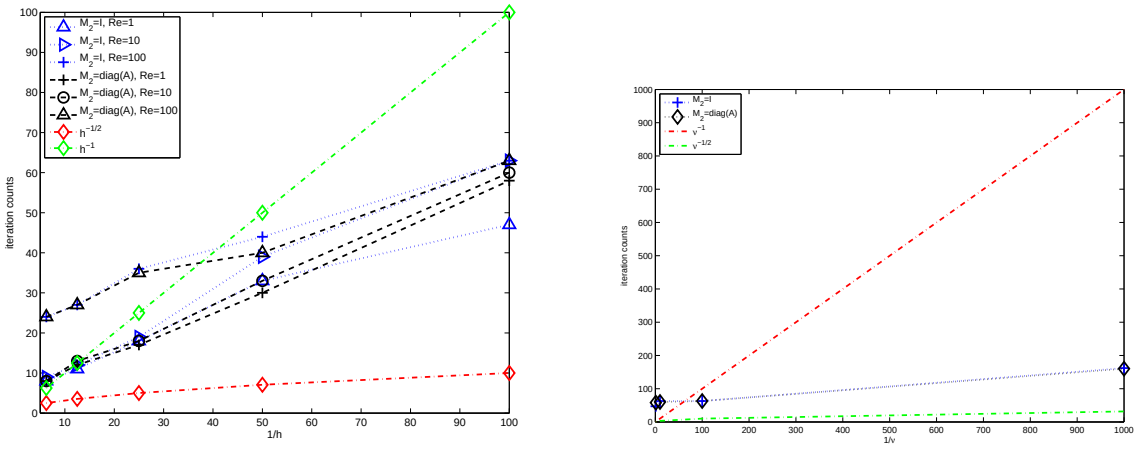
Figure 8. $\nu = 0.001$. Spectrum of the preconditioned Schur complement

$BM_2^{-1}B^T$, where M_2 can be the identity matrix I, diag(A) or the velocity mass matrix X. Since M_2 is chosen to be easily inversible (we have chosen only diagonal matrices), $BM_2^{-1}B^T$ can be computed by a sparse matricial product : the resulting matrix is sparse and its sparsity is the same as the sparsity of the classical diffusion matrix. Remark that the matrix $BM_2^{-1}B^T$ is singular with a rank equal to size$(BM_2^{-1}B^T) - 2$ for all M_2 matrices obtained by discretization of a continuous and coercive form on the velocity space. In order to obtain a non–singular matrix, we add a Dirichlet boundary condition on two degrees of freedom (since we have two meshes) corresponding to a point G_i and a point S_k.

The "real" value of the pressure is computed at the end of the solving by ensuring that the integral of the pressure is equal to 0. $BM_2^{-1}B^T\bullet = \star$ is solved using the multifrontal method from MUMPS. The product by A^{-1} is also done by a MUMPS factorization. For the BFBt preconditioner, one BICGSTAB iteration corresponds to two inversions of A and four inversions of $BM_2^{-1}B^T$ (only two inversions of $B\text{diag}(A)^{-1}B^T$ for SIMPLE). The linear iterations are stopped when the stopping criterions defined in table 1 are satisfied with

$$\text{tol} = 1e-8.$$

The maximum number of iterations is fixed to 5000 and DNC (for "Do Not Converge") means that the stopping criterion is not satisfied at the 5000th iteration. Table 2 shows the iteration count of Uzawa–BICGSTAB preconditioned by SIMPLE and BFBt preconditioners described in section 4.



(a) h-dependence (b) ν-dependence

Figure 9. Dependences for the $BFBt$ preconditioner.

Numbers of iterations for Olshanskii preconditioner have not be reported in this table because this preconditioner does not fit at all our discretization. For example, on the grid 40×40 and for $\nu = 0.1$, more than 1000 iterations are necessary to obtain the convergence.

The results for the BFBt preconditioner confirm the spectrum illustration : for large viscosity ($\nu \geq 0.1$,), choosing X as weight helps to speed-up the convergence. Then, for smaller values of the

Let $\Phi = BA^{-1}f$ and $S = BA^{-1}B^T$.
initial guess: p, u
$r = \Phi - Sp$
$\tilde{r} = r$
$\omega = 1$
$resid = \|r\|/\|\Phi\|$
for i=1, maxiter
if $\rho_1 = 0$, breakdown, EXIT, end if
if i=1 then p=r
else
$\beta = \dfrac{\rho_1}{\rho_2}\dfrac{\alpha}{\omega}$
$p = r + \beta(p - \omega v)$
end if
$\hat{p} = M^{-1}p$
$e = A^{-1}B^T\hat{p}$
$v = Be$
$\alpha = \dfrac{\rho_1}{(\tilde{r}, v)}$
$s = r - \alpha v$
if $\|s\| <$ tol $\|\Phi\|$, $\quad x = x + \alpha\hat{p}$, convergence, EXIT, end if
$\hat{s} = M^{-1}s$
$q = A^{-1}B^T\hat{s}$
$t = Bq$
$y = y + \alpha\hat{p} + \omega\hat{s}$
$x = x - \alpha e - \omega q$
$r = s - \omega t$
$\rho_2 = \rho_1$
if $\|r\| <$ tol $\|\Phi\|$, convergence, EXIT, end if
si $\omega = 0$, breakdown, EXIT, end if
end for

Table 1
pseudo-code for the preconditioned Uzawa-BICGSTAB

viscosity, diag(A) should be preferred. Indeed, we have already observed in [5] that the choice of X is clearly the best choice for the Stokes problem, whereas the preconditioner obtained with $M_2 = I$ and $M_2 = diag(A)$ are equivalent but cannot compete with $M_2 = X$. The performance results obtained for the Navier–Stokes problem with moderate Reynolds number are so a continuation of the results observed for Stokes problem. However, the discretization of the convective term used in this paper (using the Bernoulli pressure) is very different than the discretization usually chosen in the literature on the Navier-Stokes problem, in particular in the works of Elman, Silvester and Wathen [8]. We observe here that the choice of X does not fit our type of discretization for higher Reynolds number.

Figure 9 shows a mildly dependence of the viscosity which is very similar to the results of literature

mesh	preconditioner	$\nu = 1$	$\nu = 1e-1$	$\nu = 1e-2$	$\nu = 1e-3$	$\nu = 1e-4$
10×10	$M_2 = I$	8	9	24	DNC	DNC
	$M_2 = X$	7	8	29	627	DNC
	$M_2 = \text{diag}(A)$	8	8	24	DNC	DNC
	SIMPLE	47	46	62	620	DNC
20×20	$M_2 = I$	11	12	27	335	DNC
	$M_2 = X$	10	12	32	602	DNC
	$M_2 = \text{diag}(A)$	12	13	27	169	DNC
	SIMPLE	87	93	92	288	DNC
40×40	$M_2 = I$	18	19	36	211	DNC
	$M_2 = X$	15	15	42	DNC	DNC
	$M_2 = \text{diag}(A)$	17	18	35	205	DNC
	SIMPLE	154	166	179	282	DNC
80×80	$M_2 = I$	33	38	44	191	DNC
	$M_2 = X$	19	22	72	DNC	DNC
	$M_2 = \text{diag}(A)$	30	33	40	170	DNC
	SIMPLE	310	370	376	394	DNC
160×160	$M_2 = I$	47	63	63	162	DNC
	$M_2 = X$	27	30	96	DNC	DNC
	$M_2 = \text{diag}(A)$	58	60	63	160	DNC
	SIMPLE	685	881	909	871	

Table 2
Iteration count for the linear solver

[19,9]: indeed, the independence is only obtained for constant wind. Since we have a circulating flow, the h dependence is more important but of the same order than for the results obtained by Olshanskii and Vassilevski [19].

6 Conclusion

In this work, we have shown that Elman's BFBt preconditioner is well adapted to the solution of Oseen problems with this finite volume discretization and allows to decrease considerably the

iteration count; this is not the case for all classical preconditioners developed for finite element method problems, as pointed out in our numerical results. The algebraic framework of the preconditioner makes it easy to implement to general discretizations (finite elements, finite volumes, ...), so it can be presented as a robust preconditioner for saddle–point problems. It can be efficiently coupled with nonlinear iterations for solving Navier-Stokes equations, at least for moderate Reynolds numbers ($Re \simeq 1000$).

References

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

[2] J. Cahouet and J.P Chabard. Some fast 3D finite element solvers for the generalized Stokes problem. *Int. J. Numer. Meth. Fluids.*, 8:869–895, 1988.

[3] C. Calgaro, P. Deuring and D. Jennequin. A preconditioner for generalized saddle–point problems: Application to 3D stationary Navier–Stokes equations. *Numer. Meth. Part. Diff. Eq.*, 22(6):1289-1313, 2006.

[4] S. Delcourte, K. Domelevo and P. Omnes. A discrete duality finite volume approach to hodge decomposition and div-curl problems on almost arbitrary two-dimensional meshes. *SIAM J. Numer. Anal.*, 45(3):1142-1174, 2007.

[5] S. Delcourte. *Développement de méthodes de volumes finis pour la mécanique des fluides.* PhD thesis, Université Paul Sabatier, Toulouse III, 2007 (available at http://tel.archives-ouvertes.fr/tel-00200833/).

[6] S. Delcourte and D. Jennequin. Preconditioning Navier–Stokes problem Discretized by Discrete Duality Finite Volume schemes. In R. Eymard and J.-M. Hérard, editors, *Proceedings Finite Volumes for Complex Applications V*, 329–336, 2008.

[7] K. Domelevo and P. Omnes. A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids. *ESAIM: M2AN*, 39(6):1203–1249, 2005.

[8] Howard C. Elman, David J. Silvester and Andrew J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, 2005,

[9] H. Elman, V.E. Howle, J. Shadid, R. Shuttleworth and R. Tuminaro. Block preconditioners based on approximate commutators. *SIAM J. Sci. Comput.*, 27(5):1651–1668 (electronic), 2006.

[10] H. Elman. Preconditioning for the steady-state Navier-Stokes equations with low viscosity. *SIAM J. Sci. Comput.*, 20:1299–1316, 1999.

[11] EMC2 Web Page. "http://www-rocq.inria.fr/modulef/".

[12] H.C. Elman and D. Silvester. Fast nonsymmetric iterations and preconditioning for Navier-Stokes equations. *SIAM J. Sci. Comput.*, 17:33–46, 1996.

[13] D. Kay and D. Loghin. A Green's function preconditioner for the steady–state Navier-Stokes equations. Technical Report 99/06, Oxford University Computing Laboratory, 1999.

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

[15] A. Klawonn and G. Starke. Block triangular preconditioners for nonsymmetric saddle point problems: field-of-values analysis. *Numer. Math.*, 81:577–594, 1999.

[16] G.H. Golub, M. Benzi and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 2005.

[17] MUMPS Web Page. "http://mumps.enseeiht.fr/index.html".

[18] M.A. Olshanskii. A low order finite element method for the Navier-Stokes equations of steady incompressible flow: a stabilization issue and iterative methods. *Comput. Methods Appl. Mech. Eng.*, 191:5515–5536, 2002.

[19] M.A. Olshanskii and Y.V. Vassilevski. Pressure Schur complement preconditioners for the discrete Oseen problem. *SIAM J. Sci. Comput.*, 29(6):2686–2704 (electronic), 2007.

[20] D. Silvester, H. Elman, D. Kay and A. Wathen. Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow. *J. Comput. Appl. Math.*, 128:261–279, 2001.

[21] D. Silvester and A. Wathen. Fast iterative solution of stabilised Stokes systems II: Using general block preconditioners. *SIAM J. Numer. Anal.*, 31:1352–1367, 1994.

[22] S. Turek. *Efficient solvers for incompressible flow problems*, Vol. 6 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 1999.

[23] H.A. Van Der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 13(2):631–644, 1992.