A sixth-order finite volume scheme for the steady-state incompressible Stokes equations on staggered unstructured meshes

Ricardo Costa, Stéphane Clain, Gaspar Machado

A sixth-order finite volume scheme for the steady-state incompressible Stokes equations on staggered unstructured meshes

Ricardo Costa[a], Stéphane Clain[a], Gaspar J. Machado[a]

[a] *Centre of Mathematics, University of Minho, Campus of Azurém, 4800-058 Guimarães, Portugal*
e-mail: ricardodpcosta@gmail.com, {clain,gjm}@math.uminho.pt

Abstract

We propose a new sixth-order finite volume scheme to solve the bidimensional linear steady-state Stokes problem on staggered unstructured meshes and complex geometries. The method is based on several classes of polynomial reconstructions to accurately evaluate the diffusive fluxes, the pressure gradient, and the velocity divergence. The main difficulty is to handle the div-grad duality to avoid numerical locking and oscillations. A new preconditioning technique based on the construction of a pseudo-inverse matrix is also proposed to dramatically reduce the computational effort. Several numerical simulations are carried out to highlight the performance of the new method.

Keywords: Stokes equations, incompressible fluid, finite volume, high-order scheme, preconditioning

1. Introduction

The Stokes problem together with the Darcy system are two classical prototype models of mixed problems where the pressure function derives from the divergence-free velocity constraint. The finite volume method turns to be a natural framework to design built-in conservation schemes since the pioneer book of Patankar [39] and we refer to the textbook of Ferziger and Peric [17] for an overview of the finite volume for the Navier-Stokes equations. A large range of schemes has been developed to provide both accurate and stable solutions where one can distinguish different kinds of approach namely staggered or collocated discretizations, structured grids or unstructured meshes for complex geometries, and coupled or segregated velocity and pressure leading to a saddle point problem or a projection method in the divergence-free space (see the introductions of [22, 44] for a short overview). Another fundamental challenge concerns the preconditioning of the linear system deriving from the space discretization.

Second-order methods are a standard in industry for the computation of incompressible flow and in commercial software development. There exists an important literature and books on the subject using the finite difference [8, 41], the finite element [43, 52, 7, 23, 54], or the finite volume approach [39, 20, 53, 49, 15, 5]. However, there are fewer papers on higher order approximations (greater than the second-order of accuracy). Very high-order schemes for incompressible fluid flow have been developed

using the finite difference framework using the Padé methodology (the so-called compact scheme) [25] on staggered structured grids (see [29, 22] and references herein) for the fourth-order case, and more recently for the sixth-order case [4]. Finite element [23, 19] and discontinuous Galerkin methods [34, 16, 36] also received important contributions to achieve very high-order approximation both in time and space.

Accurate finite volume approximations are receiving considerable attention to compute approximations for the Euler system and the compressible Navier-Stokes equations in two- and three-dimensional geometries [38, 33, 28, 12] but the incompressible case is far from being so well-developed. Pereira *et al.* [40] and Smirnov *et al.* [51] proposed a fourth-order finite volume method on structured grids based on the Padé technique and several very high-order schemes have been developed involving compact stencils to provide accurate approximations [42, 26, 31, 18]. The major drawback of the compact technology derives from the restriction to structured grids since it turns to be very complicated for unstructured meshes. Up to our knowledge, the use of very high-order finite volume methods for the (Navier)-Stokes problem with unstructured meshes has only been tackled in [37, 44] introducing fourth-order and sixth-order schemes based on the application of a mesh-free technique (Moving Least Squares refereed as FV-MLS method) to the finite volume framework.

In the present paper we propose a new sixth-order finite volume method for the steady-state incompressible Stokes equations with unstructured meshes based on the technology initially developed for the convection-diffusion problem in [9, 6]. We use a staggered discretization with a primal unstructured mesh for the pressure and the associated diamond mesh for the velocity to avoid the Rhie-Chow interpolation [46, 49]. The coupled velocity-pressure approach is employed to avoid the pressure correction intermediate step to provide the divergence-free velocity [22]. Moreover, we do not treat the steady-state as the asymptotic limit of an artificial time marching problem but we directly solve the linear system associated to the saddle point problem. The main difficulty is to achieve an efficient approximation of the solution taking into account the divergence-free velocity constraint to determine the pressure. The method is based, on the one hand, in different kinds of polynomial reconstructions to compute the viscous flux, the pressure gradient, and the velocity divergence up to a sixth-order of accuracy and, on the other hand, in a matrix-free formulation using the residual method as in [9, 6]. Unlike the popular second-order methods, the underlying global matrix is not symmetric hence the classical argument to guarantee the existence of divergence-free numerical approximations does not hold any longer. Nevertheless, we show that the method enables to carry out accurate approximations and the algebraic solver (here GMRES) converges to the steady-state solution. A new kind of preconditioning matrix is also given based on the original method proposed in [9]. The preconditioning technique has been adapted to the specific diamond structure of the dual mesh and we numerically prove the efficiency of our preconditioner.

The document is divided in seven sections. After the introduction, we present in Section 2 all the geometrical ingredients and notations we need to develop the generic finite volume scheme. We detail in Section 3 the different types of polynomial reconstructions to achieve local sixth-order representations of the underlying solution, the numerical fluxes and the numerical scheme and in Section 4 we present the new preconditioning procedure. In Section 5 we address the numerical assessments to show the robustness and accuracy of the proposed techniques and in Section 6 we present the simulation of a

polymer extruder apparatus. We end the document with some conclusions.

2. Finite volume for the Stokes equations

Let Ω be an open bounded polygonal domain of $\mathbb{R}^2$ with boundary $\partial\Omega$ and $x = (x_1, x_2)$. We seek functions $U = (U_1, U_2) \equiv (U_1(x), U_2(x))$, the velocity field, and $P \equiv P(x)$, the pressure, solutions of the steady-state flow of an incompressible Newtonian fluid governed by the Stokes equations

$$\nabla \cdot (-\mu \nabla U + P I_2) = f, \qquad \text{in } \Omega, \tag{1}$$

$$\nabla \cdot U = 0, \qquad \text{in } \Omega, \tag{2}$$

where the dynamic viscosity $\mu \equiv \mu(x)$ and the source term $f = (f_1, f_2) \equiv (f_1(x), f_2(x))$ are given regular functions. The tensor ∇U is defined as $[\nabla U]_{\alpha\beta} = \frac{\partial U_\alpha}{\partial x_\beta}$, $\alpha, \beta = 1, 2$, and I_2 stands for the identity matrix in $\mathbb{R}^{2\times 2}$. The system (1-2) is completed with the Dirichlet boundary condition

$$U = U_{\mathrm{D}}, \qquad \text{on } \partial\Omega, \tag{3}$$

where $U_{\mathrm{D}} = (U_{1,\mathrm{D}}, U_{2,\mathrm{D}}) \equiv (U_{1,\mathrm{D}}(x), U_{2,\mathrm{D}}(x))$ is a given regular function on $\partial\Omega$ which satisfies the compatibility condition

$$\int_{\partial\Omega} U_{\mathrm{D}} \cdot n \, \mathrm{d}s = 0,$$

with $n = (n_1, n_2)$ the outward unit normal vector on $\partial\Omega$. Moreover, uniqueness for the pressure is guaranteed by the additional constraint $\int_\Omega P \, \mathrm{d}x = 0$.

2.1. Primal and diamond meshes

The primal mesh of Ω, that we denote by $\mathcal{M}$, is a partition of Ω into I non-overlapping convex polygonal cells c_i, $i \in \mathcal{C}_\mathcal{M} = \{1, \ldots, I\}$, and adopt the notations we detail hereafter (see Fig. 1, left):

- for any cell c_i, $i \in \mathcal{C}_\mathcal{M}$, we denote by ∂c_i its boundary and by $|c_i|$ its area; the reference cell point is denoted by m_i which can be any point in c_i (in the present work we shall consider the centroid);
- two cells c_i and c_j share a common edge e_{ij} whose length is denoted by $|e_{ij}|$ and $n_{ij} = (n_{1,ij}, n_{2,ij})$ is the unit normal vector to e_{ij} outward to c_i, *i.e.* $n_{ij} = -n_{ji}$; the reference edge point is m_{ij} which can be any point in e_{ij} (in the present work we consider the midpoint); if an edge of c_i belongs to the boundary, the index j is tagged by the letter D;
- for any cell c_i, $i \in \mathcal{C}_\mathcal{M}$, we associate the index set $\nu(i) \subset \{1, \cdots, I\} \cup \{\mathrm{D}\}$ such that $j \in \nu(i)$ if e_{ij} is a common edge of cells c_i and c_j or with the boundary if $j = \mathrm{D}$.

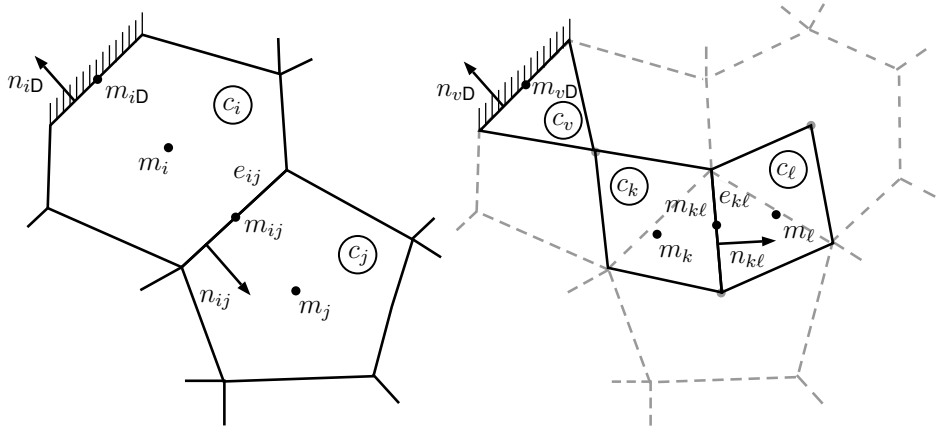

Figure 1: Notation for the primal mesh (left) and for the diamond mesh (right).

The diamond mesh of Ω, that we denote by $\mathcal{D}$, derives from the primal mesh $\mathcal{M}$ and is constituted of K non-overlapping diamond-shape cell (which degenerate to triangles in the boundary) c_k, $k \in \mathcal{C}_\mathcal{D} = \{I+1, \ldots, I+K\}$. Indeed, for each inner primal edge e_{ij} corresponds a unique cell of the diamond mesh defined by the reference points m_i and m_j and the vertices of the edge (the dual cell associated to a boundary edge $e_{i\mathrm{D}}$ is defined by the reference point m_i and the vertices of the edge).

The notation for the diamond mesh follows the notation introduced for the primal mesh where we substitute the index $i \in \mathcal{C}_\mathcal{M}$ by $k \in \mathcal{C}_\mathcal{D}$ and the index $j \in \nu(i)$ by $\ell \in \nu(k)$ (see Fig. 1, right). In particular m_k is any point in c_k (in the present work we shall consider the centroid) and $m_{k\ell}$ is any point in $e_{k\ell}$ (in the present work we consider the midpoint).

To define the association between diamond cells and primal edges, we introduce the correspondence operator $\Pi_\mathcal{D}$ such that for given arguments (i,j), $i \in \mathcal{C}_\mathcal{M}$, $j \in \nu(i)$, we associate the corresponding diamond cell index $k = \Pi_\mathcal{D}(i,j) \in \mathcal{C}_\mathcal{D}$. In the same way, for each diamond edge, we introduce the correspondence operator $\Pi_\mathcal{M}$ such that for given arguments (k,ℓ), $k \in \mathcal{C}_\mathcal{D}$, $\ell \in \nu(k)$, we associate the corresponding primal cell index $i = \Pi_\mathcal{M}(k,\ell) \in \mathcal{C}_\mathcal{M}$.

The numerical integrations on the edges are performed with Gaussian quadrature where for the primal edges e_{ij}, $i \in \mathcal{C}_\mathcal{M}$, $j \in \nu(i)$, we denote by $q_{ij,r}$, $r = 1, \ldots, R$, their Gauss points and for the diamond edges $e_{k\ell}$, $k \in \mathcal{C}_\mathcal{D}$, $\ell \in \nu(k)$, we denote by $q_{k\ell,r}$, $r = 1, \ldots, R$, their Gauss points, both sets with weights ζ_r, $r = 1, \ldots, R$ (see Fig. 2).

2.2. Generic finite volume scheme

To provide the generic very high-order finite volume scheme, we first integrate equation (1) over each diamond cell c_k, $k \in \mathcal{C}_\mathcal{D}$, and then apply the divergence theorem, yielding

$$\int_{\partial c_k} (-\mu \nabla U + P I_2)\, n \, \mathrm{d}s = \int_{c_k} f \, \mathrm{d}x,$$

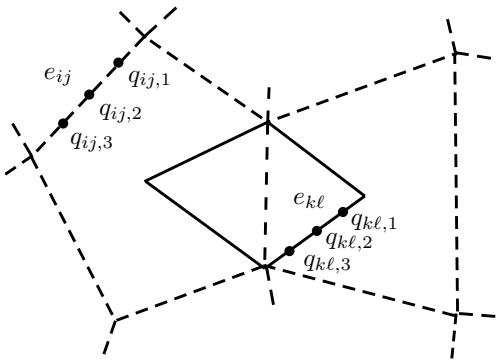

Figure 2: Gauss points on the edges of the primal cells (dashes lines) and on the edges of the diamond cells (solid lines).

which can be rewritten in the scalar form as

$$\int_{\partial c_k} (-\mu \nabla U_\beta \cdot n + P n_\beta) \ \mathrm{d}s = \int_{c_k} f_\beta \ \mathrm{d}x, \ \beta = 1, 2.$$

Considering the Gaussian quadrature with R points, *i.e.* of order $2R$, for the line integrals, we get the residual expression

$$\sum_{\ell \in \nu(k)} \frac{|e_{k\ell}|}{|c_k|} \left[\sum_{r=1}^{R} \zeta_r \left(\mathbb{F}^U_{\beta,k\ell,r} + \mathbb{F}^P_{\beta,k\ell,r} \right) \right] - f_{\beta,k} = \mathcal{O}\left(h_k^{2R} \right), \ \beta = 1, 2, \tag{4}$$

with the physical fluxes given by

$$\mathbb{F}^U_{\beta,k\ell,r} = -\mu(q_{k\ell,r}) \nabla U_\beta(q_{k\ell,r}) \cdot n_{k\ell}, \quad \mathbb{F}^P_{\beta,k\ell,r} = P(q_{k\ell,r}) n_{\beta,k\ell},$$

$h_k = \max_{\ell \in \nu(k)} |e_{k\ell}|$, and $f_{\beta,k}$ an approximation of order $2R$ of the mean value of f_β over cell c_k (if cell c_k is not triangular, we split it into sub-triangles which share the cell centroid as a common vertex and apply the quadrature rule on each sub-triangle as in [13]).

We now integrate equation (2) over each primal cell c_i and apply again the divergence theorem, yielding

$$\int_{\partial c_i} U \cdot n \ \mathrm{d}s = 0.$$

Considering again Gaussian quadrature with R points for the line integrals, we get the residual expression

$$\sum_{j \in \nu(i)} \frac{|e_{ij}|}{|c_i|} \sum_{r=1}^{R} \zeta_r \, \mathbb{F}^\nabla_{ij,r} = O(h_i^{2R}), \tag{5}$$

with the physical flux given by

$$\mathbb{F}^\nabla_{ij,r} = U(q_{ij,r}) \cdot n_{ij}$$

and $h_i = \max_{j \in \nu(i)} |e_{ij}|$.

3. Polynomial reconstructions and high-order finite volume scheme

The polynomial reconstruction is a powerful tool to provide an accurate local representation of the underlying solution and was initially introduced in [1, 2] for hyperbolic problems. In [9] a new methodology was proposed in the context of convection-diffusion problems in order to achieve very accurate approximations of the gradient fluxes and to take into account the boundary conditions. The authors introduced different types of polynomial reconstructions namely the conservative reconstruction in cells and on Dirichlet boundary edges and the non-conservative reconstruction on inner edges, in order to compute approximations of the convective and the diffusive fluxes. We now adapt this technology for the specific Stokes problem where the main difficulty is to handle the two meshes.

3.1. Stencil and data

A stencil is a collection of cells situated in the vicinity of a reference geometrical entity, namely an edge or a cell where the number of elements of the stencil shall depend on the degree d of the polynomial function we intend to construct. For each diamond edge $e_{k\ell}$, $k \in \mathcal{C}_\mathcal{D}$, $\ell \in \nu(k)$, we associate the stencil $S_{k\ell}$ consisting of the indices of neighbor diamond cells. Analogously, we associate the stencil S_k for each diamond cell c_k, $k \in \mathcal{C}_\mathcal{D}$, and the stencil S_i for each primal cell c_i, $i \in \mathcal{C}_\mathcal{M}$, consisting of the indices of neighbor dual and primal cells, respectively.

Remark 1. *A polynomial reconstruction of degree d requires $n_d = (d+1)(d+2)/2$ coefficients. So, in practice, a stencil consists of the N_d closest cells to each geometrical entity (edge or cell) in the respective mesh, with $N_d \geqslant n_d$ (we consider $N_d \approx 1.5 n_d$ for the sake of robustness).*

Now, we want to compute the polynomial reconstructions based on the data of the associated stencil. To this end, we assume that vectors $\mathbb{U}_1 = (U_{1,k})_{k=I+1,\dots,I+K}$, $\mathbb{U}_2 = (U_{2,k})_{k=I+1,\dots,I+K}$, and $\mathbb{P} = (P_i)_{i=1,\dots,I}$ gather the approximations of the mean values of U_1 and U_2 over the diamond cells and P over the primal cells, *i.e.*

$$U_{1,k} \approx \frac{1}{|c_k|}\int_{c_k} U_1 \,\mathrm{d}x, \quad U_{2,k} \approx \frac{1}{|c_k|}\int_{c_k} U_2 \,\mathrm{d}x, \quad P_i \approx \frac{1}{|c_i|}\int_{c_i} P \,\mathrm{d}x.$$

3.2. Conservative reconstruction for primal cells

For each primal cell c_i, $i \in \mathcal{C}_\mathcal{M}$, the local polynomial approximation of the underlying solution P based on vector $\mathbb{P}$ of degree d is defined as

$$\boldsymbol{P}_i(x) = P_i + \sum_{1 \leqslant |\alpha| \leqslant d} \mathcal{R}_i^\alpha \left[(x - m_i)^\alpha - M_i^\alpha\right],$$

where $\alpha = (\alpha_1, \alpha_2)$ with $|\alpha| = \alpha_1 + \alpha_2$ and the convention $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2}$, vector $\mathcal{R}_i = (\mathcal{R}_i^\alpha)_{1 \leqslant |\alpha| \leqslant d}$ gathers the polynomial coefficients, and $M_i^\alpha = \frac{1}{|c_i|}\int_{c_i} (x - m_i)^\alpha \,\mathrm{d}x$ in order to guarantee the conservation property

$$\frac{1}{|c_i|}\int_{c_i} \boldsymbol{P}_i(x) \,\mathrm{d}x = P_i.$$

For a given stencil S_i, we consider the quadratic functional

$$E_i(\mathcal{R}_i) = \sum_{q \in S_i} \left[\frac{1}{|c_q|} \int_{c_q} \boldsymbol{P}_i(x) \, \mathrm{d}x - P_q \right]^2. \tag{6}$$

We denote by $\widehat{\mathcal{R}}_i$ the unique vector which minimizes the quadratic functional (6) and we set $\widehat{\boldsymbol{P}}_i(x)$ the polynomial which corresponds to the best approximation in the least squares sense.

3.3. Conservative reconstruction for diamond cells

For each diamond cell c_k, $k \in \mathcal{C}_\mathcal{D}$, the local polynomial approximation of the underlying functions U_1 and U_2 based on vectors $\mathbb{U}_1$ and $\mathbb{U}_2$ of degree d are defined as

$$\boldsymbol{U}_{\beta,k}(x) = U_{\beta,k} + \sum_{1 \leqslant |\alpha| \leqslant d} \mathcal{R}^\alpha_{\beta,k} \left[(x - m_k)^\alpha - M^\alpha_k \right], \quad \beta = 1, 2,$$

where vector $\mathcal{R}_{\beta,k} = (\mathcal{R}^\alpha_{\beta,k})_{1 \leqslant |\alpha| \leqslant d}$ gathers the polynomial coefficients and $M^\alpha_k = \frac{1}{|c_k|} \int_{c_k} (x - m_k)^\alpha \, \mathrm{d}x$ in order to guarantee the conservation property

$$\frac{1}{|c_k|} \int_{c_k} \boldsymbol{U}_{\beta,k}(x) \, \mathrm{d}x = U_{\beta,k}.$$

For a given stencil S_k, we consider the quadratic functional

$$E_{\beta,k}(\mathcal{R}_{\beta,k}) = \sum_{q \in S_k} \left[\frac{1}{|c_q|} \int_{c_q} \boldsymbol{U}_{\beta,k}(x) \, \mathrm{d}x - U_{\beta,q} \right]^2. \tag{7}$$

We denote by $\widehat{\mathcal{R}}_{\beta,k}$ the unique vector which minimizes the quadratic functional (7) and we set $\widehat{\boldsymbol{U}}_{\beta,k}(x)$ the polynomial which corresponds to the best approximation in the least squares sense.

3.4. Non-conservative reconstruction for inner diamond edges

For each inner diamond edge $e_{k\ell}$, $k \in \mathcal{C}_\mathcal{D}$, $\ell \in \nu(k)$, the local polynomial approximations of degree d of the underlying functions U_1 and U_2 are defined as

$$\boldsymbol{U}_{\beta,k\ell}(x) = \sum_{0 \leqslant |\alpha| \leqslant d} \mathcal{R}^\alpha_{\beta,k\ell} (x - m_{k\ell})^\alpha, \quad \beta = 1, 2,$$

where vector $\mathcal{R}_{\beta,k\ell} = (\mathcal{R}^\alpha_{\beta,k\ell})_{0 \leqslant |\alpha| \leqslant d}$ gathers the polynomial coefficients (notice that in this case $|\alpha|$ starts with 0 since no conservation property is required). For a given stencil $S_{k\ell}$ with $\# S_{k\ell}$ elements and vector $\omega_{\beta,k\ell} = (\omega_{\beta,k\ell,q})_{q=1,\ldots,\# S_{k\ell}}$ of the positive weights of the reconstruction, we consider the quadratic functional

$$E_{\beta,k\ell}(\mathcal{R}_{\beta,k\ell}) = \sum_{q \in S_{k\ell}} \omega_{\beta,k\ell,q} \left[\frac{1}{|c_q|} \int_{c_q} \boldsymbol{U}_{\beta,k\ell}(x) \, \mathrm{d}x - U_{\beta,q} \right]^2. \tag{8}$$

We denote by $\widetilde{\mathcal{R}}_{\beta,k\ell}$ the unique vector which minimizes the quadratic functional (8) and we set $\widetilde{\boldsymbol{U}}_{\beta,k\ell}(x)$ the polynomial which corresponds to the best approximation in the least squares sense.

3.5. Conservative reconstruction for diamond boundary edges

We treat the boundary diamond edges in a particular way in order to take into account the Dirichlet boundary conditions prescribed for the velocity. For each boundary diamond edge $e_{k\mathrm{D}}$, $k \in \mathcal{C}_{\mathcal{D}}$, the local polynomial approximations of degree d of the underlying functions U_1 and U_2 are defined as

$$\boldsymbol{U}_{\beta,k\mathrm{D}}(x) = U_{\beta,k\mathrm{D}} + \sum_{1\leqslant|\alpha|\leqslant d} \mathcal{R}^{\alpha}_{\beta,k\mathrm{D}} \left[(x - m_{k\mathrm{D}})^{\alpha} - M^{\alpha}_{k\mathrm{D}}\right], \quad \beta = 1, 2,$$

where vector $R_{\beta,k\mathrm{D}} = (R^{\alpha}_{\beta,k\mathrm{D}})_{1\leqslant|\alpha|\leqslant d}$ gathers the polynomial coefficients, $U_{\beta,k\mathrm{D}}$ is an approximation of the mean value $U_{\beta,\mathrm{D}}$ of order $2R$ over the diamond boundary edge $e_{k\mathrm{D}}$, and $M^{\alpha}_{k\mathrm{D}} = \frac{1}{|e_{k\mathrm{D}}|}\int_{e_{k\mathrm{D}}}(x - m_{k\mathrm{D}})^{\alpha}\,\mathrm{d}x$ in order to guarantee the conservation property

$$\frac{1}{|e_{k\mathrm{D}}|}\int_{e_{k\mathrm{D}}} \boldsymbol{U}_{\beta,k\mathrm{D}}(x)\,\mathrm{d}s = U_{\beta,k\mathrm{D}}.$$

For a given stencil $S_{k\mathrm{D}}$ with $\#S_{k\mathrm{D}}$ elements and vector $\omega_{\beta,k\mathrm{D}} = (\omega_{\beta,k\mathrm{D},q})_{q=1,\ldots,\#S_{k\mathrm{D}}}$ of the positive weights of the reconstruction, we consider the quadratic functional

$$E_{\beta,k\mathrm{D}}(\mathcal{R}_{\beta,k\mathrm{D}}) = \sum_{q\in S_{k\mathrm{D}}} \omega_{\beta,k\mathrm{D},q}\left[\frac{1}{|c_q|}\int_{c_q} \boldsymbol{U}_{\beta,k\mathrm{D}}(x)\,\mathrm{d}x - U_{\beta,q}\right]^2. \tag{9}$$

We denote by $\widehat{\mathcal{R}}_{\beta,k\mathrm{D}}$ the unique vector which minimizes the quadratic functional (9) and we set $\widehat{\boldsymbol{U}}_{\beta,k\mathrm{D}}(x)$ the polynomial which corresponds to the best approximation in the least squares sense.

Remark 2. *The motivation for introducing the weights in the case of a non-conservative polynomial reconstruction and in the case of a conservative polynomial reconstruction for Dirichlet boundary edges, is presented in [9] as well as the importance to set larger values for the adjacent cells.*

3.6. High-order finite volume scheme

This subsection is dedicated to design high-order numerical flux approximations based on the polynomial reconstructions presented in the previous subsections to provide the global residual operator.

3.6.1. Numerical fluxes

For a given polynomial degree d and the associated stencils which guarantee the d-consistency property (see [9]), four numerical fluxes situations arise:

- for an inner diamond edge $e_{k\ell}$, the fluxes at the quadrature point $q_{k\ell,r}$ write

  $$\mathcal{F}^{U}_{\beta,k\ell,r} = -\mu(q_{k\ell,r})\nabla\widetilde{\boldsymbol{U}}_{\beta,k\ell}(q_{k\ell,r})\cdot n_{k\ell} \quad \text{and} \quad \mathcal{F}^{P}_{\beta,k\ell,r} = \widehat{\boldsymbol{P}}_{i}(q_{k\ell,r})n_{\beta,k\ell}, \quad \beta = 1, 2,$$

  with the correspondence $i = \Pi_{\mathcal{M}}(k, \ell)$;

- for a boundary diamond edge $e_{k\mathrm{D}}$, the fluxes at the quadrature point $q_{k\mathrm{D},r}$ write

$$\mathcal{F}^U_{\beta,k\mathrm{D},r} = -\mu(q_{k\mathrm{D},r})\nabla\widehat{\boldsymbol{U}}_{\beta,k\mathrm{D}}(q_{k\mathrm{D},r})\cdot n_{k\mathrm{D}} \quad \text{and} \quad \mathcal{F}^P_{\beta,k\mathrm{D},r} = \widehat{\boldsymbol{P}}_i(q_{k\mathrm{D},r})n_{\beta,k\mathrm{D}}, \ \ \beta = 1,2,$$

with the correspondence $i = \Pi_{\mathcal{M}}(k,\mathrm{D})$;

- for an inner primal edge e_{ij}, the flux at the quadrature point $q_{ij,r}$ writes

$$\mathcal{F}^\nabla_{ij,r} = \widehat{\boldsymbol{U}}_{1,k}(q_{ij,r})n_{1,ij} + \widehat{\boldsymbol{U}}_{2,k}(q_{ij,r})n_{2,ij},$$

with the correspondence $k = \Pi_{\mathcal{D}}(i,j)$;

- for a boundary primal edge $e_{i\mathrm{D}}$, the flux at the quadrature point $q_{i\mathrm{D},r}$ writes

$$\mathcal{F}^\nabla_{i\mathrm{D},r} = U_{1,\mathrm{D}}(q_{i\mathrm{D},r})n_{1,i\mathrm{D}} + U_{2,\mathrm{D}}(q_{i\mathrm{D},r})n_{2,i\mathrm{D}}.$$

3.6.2. Residual operators

For any vector $\Phi = (\mathbb{U}_1, \mathbb{U}_2, \mathbb{P})$ in $\mathbb{R}^{2K+I}$, we define the residual operators for each diamond cell c_k, $k \in \mathcal{C}_D$, as

$$\mathcal{G}^\beta_k(\Phi) = \sum_{\ell\in\nu(k)} \frac{|e_{k\ell}|}{|c_k|}\left[\sum_{r=1}^R \zeta_r\left(\mathcal{F}^U_{\beta,k\ell,r} + \mathcal{F}^P_{\beta,k\ell,r}\right)\right] - f_{\beta,k}, \ \ \beta = 1,2,$$

and for each primal cell c_i, $i \in \mathcal{C}_\mathcal{P}$, as

$$\mathcal{G}^\nabla_i(\Phi) = \sum_{j\in\nu(i)} \frac{|e_{ij}|}{|c_i|}\sum_{r=1}^R \zeta_r \mathcal{F}^\nabla_{ij,r},$$

which correspond to the finite volume scheme (4-5) cast in residual form. Gathering all the components of the residuals in vectors $\mathcal{G}^\beta(\Phi) = \left(\mathcal{G}^\beta_k(\Phi)\right)_{k=I+1,\ldots,I+K}$ and $\mathcal{G}^\nabla(\Phi) = \left(\mathcal{G}^\nabla_i(\Phi)\right)_{i=1,\ldots,I}$, we introduce the global affine operator from $\mathbb{R}^{2K+I}$ into $\mathbb{R}^{2K+I}$, given by

$$\mathcal{H}(\Phi) = \left(\mathcal{G}^1(\Phi), \mathcal{G}^2(\Phi), \mathcal{G}^\nabla(\Phi)\right)^T,$$

such that vector $\Phi^\star = (\mathbb{U}_1^\star, \mathbb{U}_2^\star, \mathbb{P}^\star)^T \in \mathbb{R}^{2K+I}$, solution of the problem $\mathcal{H}(\Phi) = 0$, provides a constant piecewise approximation of U_1, U_2, and P.

4. A new preconditioning technique

This section is dedicated to the development of a new efficient preconditioning method to solve the underlying affine problem.

4.1. Matrix-free problem and preconditioning

The underlying operator corresponding to the Stokes problem takes the form

$$\mathcal{H}(\Phi) = \mathcal{A}\Phi - \mathcal{B} = \begin{bmatrix} A & B^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \mathbb{U} \\ \mathbb{P} \end{bmatrix} - \begin{bmatrix} F \\ V \end{bmatrix} = 0 \tag{10}$$

where $\mathbb{U} = (\mathbb{U}_1, \mathbb{U}_2)^T \in \mathbb{R}^{2K}$ gathers the two unknown velocity components, $F = (F_1, F_2)^T \in \mathbb{R}^{2K}$ with $F_\beta = (f_{\beta,k})_{k=I+1,\dots,I+K}$, $\beta = 1, 2$, gathers the source term, and $V \in \mathbb{R}^I$ gathers the prescribed normal velocities on the boundary. Since the matrix $\mathcal{A}$ is unknown (and we do not want to explicitly assemble it), we use linear solvers which only require the residual term such as the GMRES method [48, 47].

The preconditioning of the matrix-free problem is a critical stage to dramatically reduce the computational effort and achieve accurate approximations for the affine system. Numerous techniques have been developed, namely the multigrid preconditioner [32, 27] and the SIMPLE-type preconditioner [30, 45]. Also a very popular method for preconditioning the system uses matrix

$$\mathcal{P} = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix}, \tag{11}$$

where $S = CA^{-1}B^T$ is the Schur complement of $\mathcal{A}$ with respect to A [22]. Indeed, using $\mathcal{P}$ as a preconditioning matrix theoretically provides the solution with at most four GMRES iterations as proven in [35]. Most of the preconditioning techniques have been proposed for the saddle point problem when $C = B$ corresponding to a "symmetric" discretization which preserves the duality between the divergence and the gradient operators. Such a situation arises in many first- and second-order finite volume discretizations such as the DDFV method [10, 11] or in the context of the finite element method [50]. Unfortunately, such symmetry does not hold any longer in the case of very high-order finite volume discretizations and there exists literature which provides compatibility conditions for system (10) to provide existence of a solution [3, 24].

In the present work, two major difficulties arise for applying such preconditioning technique. First, the matrices A, B, and C are not available since we consider a matrix-free problem. Second, the computation of A^{-1} and S^{-1} (or an incomplete version such as ILU) requires an important computational effort which turns unsustainable for large linear systems. To overcome these problems, we first introduce a simple and computationally fast approximation of matrices A, B, and C and then propose an efficient preconditioning technique as an extension of the one proposed in [9].

4.2. Approximations for unknown matrices A, B, and C

We introduce the approximation $\widetilde{A} \in \mathbb{R}^{2K \times 2K}$ of A using a Patankar-like discretization (also very similar to the FV4 scheme as in [14]) given by

$$\widetilde{A}(k-I, k-I) = \widetilde{A}(k+K-I, k+K-I) = \sum_{\ell \in \nu(k)} \frac{|e_{k\ell}|}{|c_k|} \frac{\mu(m_{k\ell})}{|m_k m_\ell|}, \ k \in \mathcal{C}_\mathcal{D},$$

for the diagonal entries and by

$$\widetilde{A}(k-I, \ell-I) = \widetilde{A}(k+K-I, \ell+K-I) = -\frac{|e_{k\ell}|}{|c_k|} \frac{\mu(m_{k\ell})}{|m_k m_\ell|}, \ \ k \in \mathcal{C}_\mathcal{D}, \ \ell \in \nu(k),$$

for the extra-diagonal non-null entries.

In the same way, we introduce the approximation $\widetilde{B} \in \mathbb{R}^{I\times 2K}$ of B given by (using local indices)

$$\widetilde{B}(i,k-I) = \sum_{\substack{\ell\in\nu(k)\ with\\ \Pi_{\mathcal{M}}(k,\ell)=i}} \frac{|e_{k\ell}|}{|c_k|} n_{1,k\ell},\ \ i\in\mathcal{C}_{\mathcal{M}},\ k\in\mathcal{C}_{\mathcal{D}},$$

$$\widetilde{B}(i,k+K-I) = \sum_{\substack{\ell\in\nu(k)\ with\\ \Pi_{\mathcal{M}}(k,\ell)=i}} \frac{|e_{k\ell}|}{|c_k|} n_{2,k\ell},\ \ i\in\mathcal{C}_{\mathcal{M}},\ k\in\mathcal{C}_{\mathcal{D}},$$

and the approximation $\widetilde{C} \in \mathbb{R}^{I\times 2K}$ of C given by (using local indices)

$$\widetilde{C}(i,k-I) = \frac{|e_{ij}|}{|c_i|} n_{1,ij},\ \ i\in\mathcal{C}_{\mathcal{M}},\ j\in\nu(i),\ k=\Pi_{\mathcal{D}}(i,j),$$

$$\widetilde{C}(i,k+K-I) = \frac{|e_{ij}|}{|c_i|} n_{2,ij},\ \ i\in\mathcal{C}_{\mathcal{M}},\ j\in\nu(i),\ k=\Pi_{\mathcal{D}}(i,j).$$

Note that we do not have, *a priori*, $B = C$ since B depends on the diamond mesh while C depends on the primal mesh.

4.3. Incomplete inverse matrix

In [9], we have proposed a new preconditioning technique based on the evaluation of an incomplete inverse sparse matrix. We here propose an important extension to generalize that idea and introduce a simple way to compute an approximation of $\mathcal{A}^{-1}$. Let us denote by M a square matrix in $\mathbb{R}^{n\times n}$ and denote by $\mathcal{E}_M(i)$, $i = 1,\ldots,n$, the index set of the non-null entries of row i, that is, $j \in \mathcal{E}_M(i)$ if and only if $M(i,j) \neq 0$.

Definition 1. *We say that matrix M has a symmetric structure if $\mathcal{E}_M(i) = \mathcal{E}_{M^T}(i)$, $i = 1,\ldots,n$, and two matrices M, N have the same structures if $\mathcal{E}_M(i) = \mathcal{E}_N(i)$, $i = 1,\ldots,n$.*

Definition 2. *For a given square matrix $M \in \mathbb{R}^{n\times n}$, the admissible pair (i,j), $i \in \{1,\ldots,n\}$, $j \in \mathcal{E}_M(i)$, is of type T_0 if*

$$\mathcal{E}_M(i) \cap \mathcal{E}_M(j) = \varnothing,$$

and of type T_1 if there exists a unique $k \neq i,j$ such that

$$\mathcal{E}_M(i) \cap \mathcal{E}_M(j) = k, \quad \mathcal{E}_M(i) \cap \mathcal{E}_M(k) = j, \quad \mathcal{E}_M(j) \cap \mathcal{E}_M(k) = i.$$

We say that matrix M enjoys the T_0 property (resp. T_1) if all the admissible pairs are of type T_0 (resp. T_1).

Let M and N be two square matrices in $\mathbb{R}^{n\times n}$ with the same symmetric structure, *i.e.* $\mathcal{E}(i) = \mathcal{E}_M(i) = \mathcal{E}_N(i)$, $i = 1,\ldots,n$, and let $R = NM$. One can verify the following properties:

- each diagonal element of R is given by

$$R_{ii} = N_{ii}M_{ii} + \sum_{j \in \mathcal{E}(i)} N_{ij}M_{ji}, \quad i = 1, \ldots, n; \tag{12}$$

- $\forall i = 1, \ldots, n$ and $\forall j \in \mathcal{E}(i)$, if (i, j) if of type T_0 then

$$R_{ij} = N_{ij}M_{jj} + N_{ii}M_{ij}; \tag{13}$$

- $\forall i = 1, \ldots, n$ and $\forall j \in \mathcal{E}(i)$, if (i, j) if of type T_1 then

$$R_{ij} = N_{ij}M_{jj} + N_{ik}M_{kj} + N_{ii}M_{ij}, \tag{14a}$$
$$R_{ik} = N_{ik}M_{kk} + N_{ij}M_{jk} + N_{ii}M_{ik}, \tag{14b}$$

  with $k = \mathcal{E}(i) \cap \mathcal{E}(j)$.

Of course, matrix R may contain other non-null entries than the ones associated to the admissible pairs (i, j) but the idea is to only consider such pairs to design the incomplete inverse. To this end, we have the following definition.

Definition 3. *Assume that M has a symmetric structure such that for each non-null entry M_{ij}, the pair (i, j) is of type T_0 or T_1. We say that M admits an incomplete left inverse matrix $M^\dagger$ if*

- *the matrix $M^\dagger$ has the same structure than M and*
- *let R be defined by $R_{ii} = 1$ and $R_{ij} = 0$, $i = 1, \ldots, n$, $j \in \mathcal{E}_M(i)$, then $M^\dagger M \overset{\dagger}{=} R$ where relation $\overset{\dagger}{=}$ means that we require the equality only for the entries R_{ij}, $i = 1, \ldots, n$, $j \in \{i\} \cup \mathcal{E}(i)$.*

In other words, we seek an incomplete left inverse matrix $M^\dagger$ which have the same structure than M such that the product gives the identity matrix for indices corresponding to the non-null entries of M. Such a specific structure enables to easily compute the incomplete left inverse as we present in the following theorem.

Theorem 1. *Let $M^\dagger$ be the left incomplete inverse of $M \in \mathbb{R}^{n \times n}$. Then $\forall i = 1, \ldots, n$, $\forall j \in \mathcal{E}_M(i)$:*

- *if (i, j) is of type T_0, there exists a coefficient χ_{ij} such that*

$$M^\dagger_{ij} = \chi_{ij} M^\dagger_{ii} \quad \text{with } \chi_{ij} = -\frac{M_{ij}}{M_{jj}}; \tag{15}$$

- *if (i, j) is of type T_1, there exist coefficients χ_{ij}, χ_{ik}, with $k = \mathcal{E}_M(i) \cap \mathcal{E}_M(j)$, solution of the linear system*

$$\begin{bmatrix} M_{jj} & M_{kj} \\ M_{jk} & M_{kk} \end{bmatrix} \begin{bmatrix} \chi_{ij} \\ \chi_{ik} \end{bmatrix} = - \begin{bmatrix} M_{ij} \\ M_{ik} \end{bmatrix} \tag{16}$$

  such that

$$M^\dagger_{ij} = \chi_{ij} M^\dagger_{ii}, \quad M^\dagger_{ik} = \chi_{ik} M^\dagger_{ii}. \tag{17}$$

Moreover, we have

$$M_{ii}^{\dagger} = \frac{1}{M_{ii} + \sum_{j \in \mathcal{E}_M(i)} \chi_{ij} M_{ji}}.$$

PROOF. If (i,j) is of type T_0, relation (13) yields $M_{ij}^{\dagger} M_{jj} + M_{ii}^{\dagger} M_{ij} = 0$ then $M_{ij}^{\dagger} = -\frac{M_{ij}}{M_{jj}} M_{ii}^{\dagger}$ hence $M_{ij}^{\dagger} = \chi_{ij} M_{ii}^{\dagger}$ with $\chi_{ij} = -\frac{M_{ij}}{M_{jj}}$. Assume now that (i,j) is of type T_1. Then, relation (14) yields

$$\begin{aligned} M_{ij}^{\dagger} M_{jj} + M_{ik}^{\dagger} M_{kj} + M_{ii}^{\dagger} M_{ij} &= 0, \\ M_{ik}^{\dagger} M_{kk} + M_{ij}^{\dagger} M_{jk} + M_{ii}^{\dagger} M_{ik} &= 0, \end{aligned}$$

that we rewrite as

$$\begin{bmatrix} M_{jj} & M_{kj} \\ M_{jk} & M_{kk} \end{bmatrix} \begin{bmatrix} M_{ij}^{\dagger} \\ M_{ik}^{\dagger} \end{bmatrix} = -M_{ii}^{\dagger} \begin{bmatrix} M_{ij} \\ M_{ik} \end{bmatrix}.$$

Solving the linear system (16), we provide coefficients χ_{ij}, χ_{ik} and relations (17) hold.

From relation (12), we have

$$1 = M_{ii}^{\dagger} M_{ii} + \sum_{j \in \mathcal{E}_M(i)} M_{ij}^{\dagger} M_{ji}.$$

Having the coefficients χ_{ij} in hand, we compute the diagonal entries $M_{ii}^{\dagger}$ with

$$1 = M_{ii}^{\dagger} M_{ii} + \sum_{j \in \mathcal{E}_M(i)} \chi_{ij} M_{ji} M_{ii}^{\dagger}.$$

Hence we deduce

$$M_{ii}^{\dagger} = \frac{1}{M_{ii} + \sum_{j \in \mathcal{E}_M(i)} \chi_{ij} M_{ji}}.$$

At last we compute $M_{ij}^{\dagger}$, $i = 1, \ldots, n$, $j \in \mathcal{E}_M(i)$ with equations (15) and (17). □

Definition 4. *Let N be a $\mathbb{R}^{n \times n}$ matrix and M a matrix of symmetric structure with entries of type T_0 or T_1. We say that $M^{\dagger}$ is a left incomplete inverse preconditioning matrix of N when we solve $M^{\dagger}(Nx - b) = 0$ in place of solving $Nx - b = 0$.*

Remark 3. *Notice that the usual expression "preconditioning matrix" given for example in [48] refers to M and not to M^{-1} since one solves $Nx = b$ using preconditioning technique such as ILU or diagonal matrix where M is simpler than N. In particular, we do not compute M^{-1} explicitly, except for some very elementary cases (diagonal matrices). On the other hand, we use the expression "incomplete inverse preconditioning matrix" to underline that explicitly compute $M^{\dagger}$ and perform the preconditioning multiplying $M^{\dagger}$ with N to reduce the conditioning number of the global linear system. This yields great advantages since the preconditioning procedure just requires, in practice, the product of a sparse matrix with a vector that could be efficiently parallelized.*

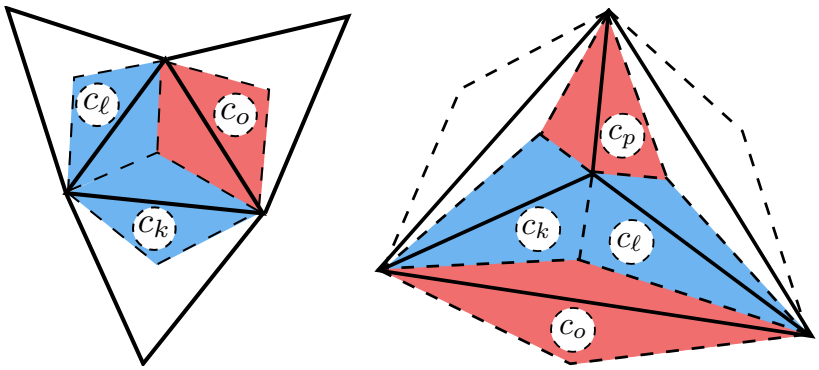

Figure 3: Correct configuration (left) for a triangular mesh where the corresponding diamond cells c_k and c_ℓ share a unique cell c_o, and undesirable configuration (right) where the diamond cells c_k and c_ℓ share two cells, c_o and c_p.

4.4. Projection operators $\mathbb{T}_\mathcal{M}$ and $\mathbb{T}_\mathcal{D}$

In the numerical section, we need to introduce two projection operators for filtering the matrices in the sense that we shall cancel some extra-diagonal entries to suit to a specific pattern deriving from the mesh connectivities. For a given primal mesh $\mathcal{M}$, we define operator $\mathbb{T}_\mathcal{M}$ on matrices $M \in \mathbb{R}^{I\times I}$ setting

$$[\mathbb{T}_\mathcal{M}(M)]_{ii} = M_{ii};\ [\mathbb{T}_\mathcal{M}(M)]_{ij} = M_{ij}, j \in \nu(i);\ [\mathbb{T}_\mathcal{M}(M)]_{ij} = 0,\ \text{otherwise.}$$

In short, the operator keeps the extra-diagonal entries belonging to the adjacent cells and cut the other entries. Notice that $\mathbb{T}_M(\mathbb{T}_\mathcal{M}(M)) = \mathbb{T}_\mathcal{M}(M)$ which justifies the term projection. Moreover, we highlight that $\nu(i)$ plays the role of $\mathcal{E}(i)$.

Theorem 2. *Assume that the primal mesh is a Delaunay triangulation with I cells such that there do not exist three triangles where each triangle is adjacent with the two others as shown in Fig. 3, right panel. Then for any matrix $M \in \mathbb{R}^{I\times I}$, associated to the triangular mesh $\mathcal{M}$, the projection result of $\mathbb{T}_\mathcal{M}(M)$ enjoys the T_0 property.*

Proof. Let c_i be a cell and $j \in \nu(i)$. If $\nu(i) \cap \nu(j) \neq \varnothing$ then there exists a cell c_q $q \in \nu(i) \cap \nu(j)$ which share an edge with the two other triangles, *i.e.* triangles c_i, c_j, c_q corresponds to the configuration of Fig. 3, left panel. Since we assume that such situation is excluded hence $q \in \nu(i) \cap \nu(j) = \varnothing$ and $\mathbb{T}_\mathcal{M}(M)$ enjoys the T_0 property. □

In the same way, for the associated diamond mesh $\mathcal{D}$ and any matrix $M \in \mathbb{R}^{K\times K}$, we define the operator $\mathbb{T}_\mathcal{D}$ by

$$[\mathbb{T}_\mathcal{D}(M)]_{kk} = M_{kk};\ [\mathbb{T}_\mathcal{D}(M)]_{k\ell} = M_{k\ell}, \ell \in \nu(k);\ [\mathbb{T}_\mathcal{D}(M)]_{k\ell} = 0,\ \text{otherwise.}$$

To deal with the vectorial case (velocity) with matrix $M \in \mathbb{R}^{2K\times 2K}$ of the form

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

we extend operator $\mathbb{T}_\mathcal{D}(M)$ setting

$$\mathbb{T}_\mathcal{D}(M) = \begin{bmatrix} \mathbb{T}_\mathcal{D}(M_{11}) & 0 \\ 0 & \mathbb{T}_\mathcal{D}(M_{22}) \end{bmatrix}.$$

Theorem 3. *Assume that a triangular primal mesh satisfies the condition of Theorem 2. Then for any matrix $M \in \mathbb{R}^{K\times K}$, the projection result of $\mathbb{T}_{\mathcal{D}}(M)$ enjoys the T_1 property.*

Proof. For any triangular primal cell we have three diamond cells, lets say c_k, c_ℓ, and c_p which share two sides between themselves, *i.e.* c_k shares an edge with c_ℓ and another edge with c_p, hence $\{c_p\} \subset \nu(k) \cap \nu(\ell)$. We claim that there is not another cell (lets say c_o) such that cells c_k and c_ℓ share a common edge with cell c_o, as shown in Fig. 3, right panel. Indeed, if there exists such a cell c_o different of c_p, then c_o contains a common edge of c_k and c_ℓ. In the present work we assume that such undesirable configuration is excluded. In conclusion, $\nu(k) \cap \nu(\ell) = \{c_p\}$ such that c_k and c_ℓ are of type T_1. □

Remark 4. *Notice that the projection result $\mathbb{T}_{\mathcal{D}}(M)$ of a matrix $M \in \mathbb{R}^{K\times K}$ is of type T_1 only if the primal mesh is a triangular mesh. On the other cases, p.e. quadrilateral meshes, matrix $\mathbb{T}_{\mathcal{D}}(M)$ is of type T_0.*

In the next section, we detail how we compute the incomplete inverse preconditioning matrix $\widetilde{\mathcal{P}}^{\dagger}$ based on the approximations $\widetilde{A}$, $\widetilde{B}$, and $\widetilde{C}$ and on the procedure we have described before.

5. Numerical results

This section is dedicated to quantitatively and qualitatively assess the robustness and accuracy of the proposed method. Time consumption and computational scalability assessments are also provided in order to prove the effectiveness of the method and its capacity to be highly parallelizable.

To perform the numerical tests, we consider a fluid with viscosity $\mu = 1$ flowing in a unit square domain $\Omega =]0,1[^2$. In order to check the implementation of the method and assess the convergence rates, we manufacture an analytical solution for the given problem setting

$$U_1(x) = \frac{1}{2}\left(1 - \cos(\pi x_1)\right)\sin(\pi x_2), \quad U_2(x) = \frac{1}{2}\sin(\pi x_1)\left(\cos(\pi x_2) - 1\right),$$
$$P(x) = \frac{1}{2}\cos\left(\frac{\pi}{2}(x_1 + x_2)\right).$$

Then, the source terms are given by

$$f_1(x) = -\frac{\pi^2}{2}\left(2\cos(\pi x_1) - 1\right)\sin(\pi x_2) - \frac{\pi}{4}\sin\left(\frac{\pi}{2}(x_1 + x_2)\right),$$
$$f_2(x) = \frac{\pi^2}{2}\sin(\pi x_1)\left(2\cos(\pi x_2) - 1\right) - \frac{\pi}{4}\sin\left(\frac{\pi}{2}(x_1 + x_2)\right).$$

Boundary conditions derive from the exact solution, namely on the top side we prescribe $U_{\mathrm{D}}(x_1, 1) = (0, -\sin(\pi x_1))$, $x_1 \in]0,1[$, while we set $U_{\mathrm{D}}(1, x_2) = (\sin(\pi x_2), 0)$, $x_2 \in]0,1[$ on the right side. For the other sides, the homogeneous Dirichlet boundary condition $U_{\mathrm{D}}(x) = (0,0)$, on $\{\partial\Omega : x_1 \neq 1, x_2 \neq 1\}$ is prescribed. We plot in Fig. 4 the isocontours of the x-component of the velocity, U_1, and of the y-component of the velocity, U_2, the isocontours of the magnitude of the velocity, $||U||$, and the isocontours of the pressure, P. In all the simulations we have carried out, the weights in functional (8) are set $\omega_{\beta,k\ell,q} = 3$, $k \in \mathcal{C}_{\mathcal{D}}$, $\ell \in \nu(k)$, $q \in S_{k\ell}$, $\beta = 1, 2$, if $e_{k\ell}$ is an edge of c_q and $\omega_{\beta,k\ell,q} = 1$, otherwise, following [9].

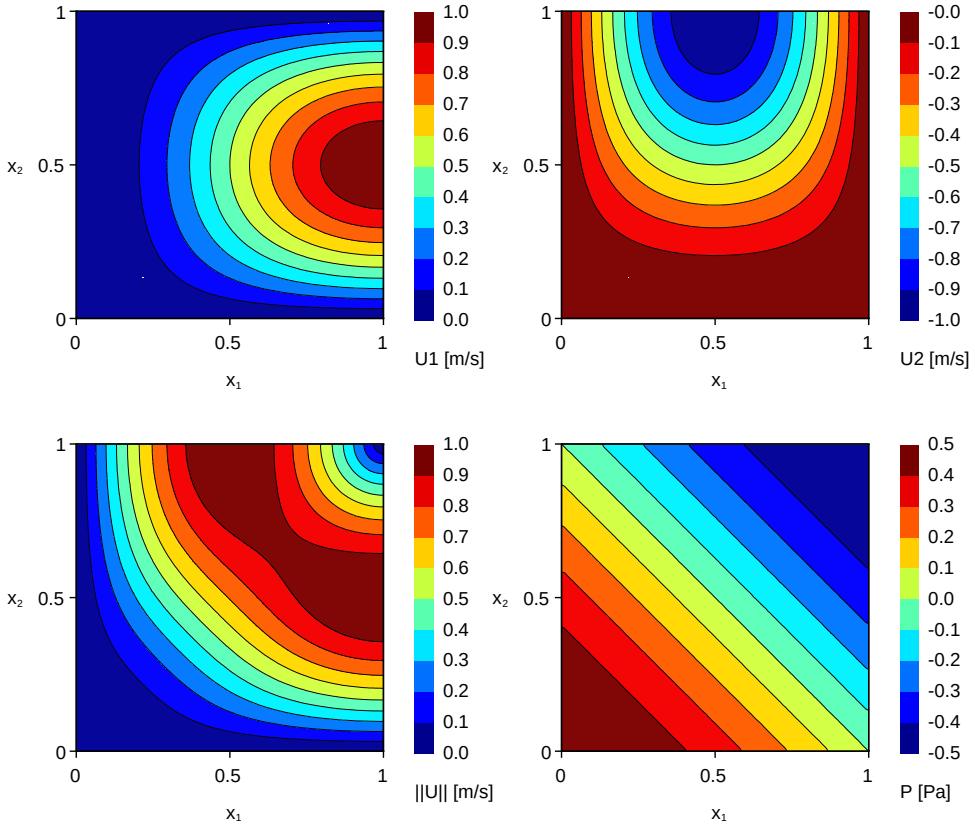

Figure 4: Isocontours of the x-component (top, left) and the y-component of the velocity (top, right), isocontours of the velocity magnitude (bottom, left), and isocontours of the pressure (bottom, right).

5.1. Accuracy and convergence rates assessment

We first assess the accuracy and the convergence rates of the numerical approximation using the manufactured solution. Vectors $\mathbb{U}^\star_\beta = (U^\star_{\beta,k})_{k\in\mathcal{C}_\mathcal{D}}$, $\beta = 1, 2$, and $\mathbb{P}^\star = (P^\star_i)_{i\in\mathcal{C}_\mathcal{M}}$ gather the approximate mean values while vectors $\overline{\mathbb{U}}_\beta = (\overline{U}_{\beta,k})_{k\in\mathcal{C}_\mathcal{D}}$, $\beta = 1, 2$, and $\overline{\mathbb{P}} = (\overline{P}_i)_{i\in\mathcal{C}_\mathcal{M}}$ gather the exact mean values of the solution given by

$$\overline{U}_{\beta,k} = \frac{1}{|c_k|}\int_{c_k} U_\beta \,\mathrm{d}x, \beta = 1, 2, \quad \text{and} \quad \overline{P}_i = \frac{1}{|c_i|}\int_{c_i} P \,\mathrm{d}x.$$

The L^1-norm errors are given by

$$E^\beta_1(\mathcal{D}) = \frac{\sum\limits_{k\in\mathcal{C}_\mathcal{D}} |U^\star_{\beta,k} - \overline{U}_{\beta,k}||c_k|}{\sum\limits_{k\in\mathcal{C}_\mathcal{D}} |c_k|}, \beta = 1, 2, \quad \text{and} \quad E^P_1(\mathcal{M}) = \frac{\sum\limits_{i\in\mathcal{C}_\mathcal{M}} |P^\star_i - \overline{P}^\star - \overline{P}_i||c_i|}{\sum\limits_{i\in\mathcal{C}_\mathcal{M}} |c_i|},$$

and the L^∞-norm errors are given by

$$E^\beta_\infty(\mathcal{D}) = \max_{k\in\mathcal{C}_\mathcal{D}} |U^\star_{\beta,k} - \overline{U}_{\beta,k}|, \beta = 1, 2, \quad \text{and} \quad E^P_\infty(\mathcal{M}) = \max_{i\in\mathcal{C}_\mathcal{M}} |P^\star_i - \overline{P}^\star - \overline{P}_i|.$$

where $\overline{P}^\star$ is the mean value of the values gather in vector $\mathbb{P}^\star$,

$$\overline{P}^\star = \frac{\sum\limits_{i\in\mathcal{C}_\mathcal{M}} P^\star_i|c_i|}{\sum\limits_{i\in\mathcal{C}_\mathcal{M}} |c_i|},$$

since the GMRES procedure does not guarantee a solution of null mean value.

Remark 5. *One can verify that the problem given by equations (1-2) is singular for the pressure since if P a solution, then $P + C$, $C \in \mathbb{R}$, is also a solution. We have consider no Dirichlet condition for the pressure (or other procedure to fix the singularity) since we have noticed that the GMRES procedure always finds the null mean value solution. However, such property does not hold for a preconditioned GMRES and therefore no convergence is achieved (the C value does not converge when we consider a finner mesh although the gradient of P is well evaluated). To overcome the problem, we simply fix $P^\star_i$ subtracting the mean value of $\mathbb{P}^\star$ such that one gets a unique piecewise approximation of P with a null mean value.*

We evaluate the convergence rate of the L^1-norm (and L^∞-norm error) between two different and successive finer primal meshes $\mathcal{M}_1$ and $\mathcal{M}_2$, with I_1 and I_2 cells, respectively, as

$$O^P_1(\mathcal{M}_1, \mathcal{M}_2) = 2\frac{|\log(E^P_1(\mathcal{M}_1)/E^P_1(\mathcal{M}_2))|}{|\log(I_1/I_2)|}.$$

In the same way, we define the convergence order between two different and successive finer diamond meshes $\mathcal{D}_1$ and $\mathcal{D}_2$ with K_1 and K_2 cells, respectively, as

$$O^\beta_1(\mathcal{D}_1, \mathcal{D}_2) = 2\frac{|\log(E^\beta_1(\mathcal{D}_1)/E^\beta_1(\mathcal{D}_2))|}{|\log(K_1/K_2)|}.$$

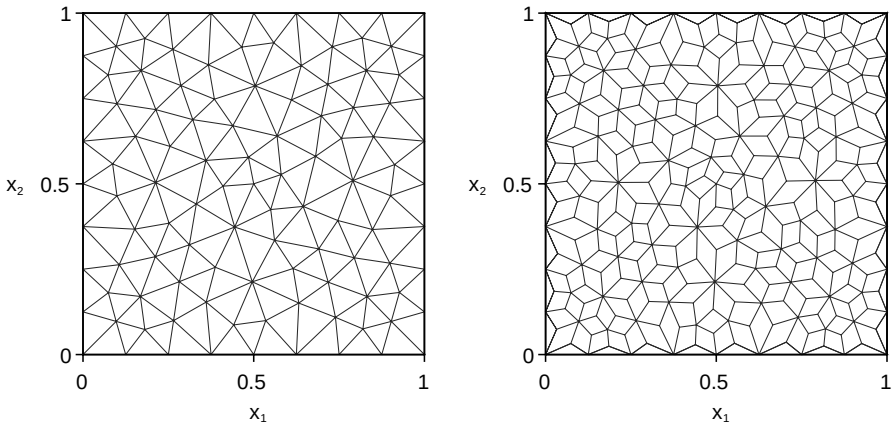

Figure 5: A coarse uniform triangular Delaunay mesh (left) and the associated diamond mesh (right).

As a first test, we carry out simulations with successive finer regular triangular Delaunay meshes (see Fig. 5, left) and the associated diamond meshes (see Fig. 5, right). We report in Tables 1, 2, and 3 the L^1- and L^∞-norm errors and the convergence rates using the $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, respectively, where the number of unknowns (the same as degrees of freedom) is $DOF = K$ for U_1 and U_2 and $DOF = I$ for P. The notation E_1 is a generalization which stands for E_1^β or E_1^P depending on the variable we are dealing with (U_β or P, respectively). The same convention is valid for E_∞, O_1, and O_∞.

The $\mathbb{P}_1$ polynomial reconstruction provides a second-order approximation for the velocity and a first-order approximation for the pressure. The scheme based on the $\mathbb{P}_3$ reconstruction achieved an effective fourth-order approximation for the velocity and a third-order (slightly better) approximation for the pressure and scheme based on the $\mathbb{P}_5$ reconstruction achieved a sixth- and fifth-order approximations for the velocity and the pressure, respectively. We also mention that no oscillations or numerical locking are reported in all the experiences.

Scheme robustness and accuracy assessments with deformed meshes are also of crucial importance in order to check the method capacity to handle complex meshes still preserving high-order convergence rates. To this end, we consider successive finer deformed quadrilateral meshes (see Fig. 6) applying a random displacement of each inner vertex of structured meshes controlled by a deformation factor (see the detailed procedure in [9]). In the present experience, we choose a test case with 30% of deformation. We report in Tables 4, 5, and 6 the L^1- and L^∞- norm errors and convergence rates using the $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, respectively (the meaning of DOF is the one just presented).

We observe that the scheme correctly handles complex meshes and the convergence rates are optimal both for the velocity and for the pressure. We achieved up to a sixth-order convergence rate for the velocity using the $\mathbb{P}_5$ polynomial reconstruction and no oscillations were noticed in all the tests.

Table 1: Errors and convergence rates for the $\mathbb{P}_1$ scheme with uniform triangular Delaunay primal meshes.

	DOF	E_1	O_1	E_∞	O_∞
U_1	363	1.23E−03	—	9.54E−03	—
	1454	3.34E−04	1.88	2.22E−03	2.10
	6135	7.96E−05	1.99	6.80E−04	1.64
	24719	1.90E−05	2.06	1.35E−04	2.33
U_2	363	1.24E−03	—	5.31E−03	—
	1454	3.42E−04	1.86	1.88E−03	1.50
	6135	7.78E−05	2.06	5.10E−04	1.81
	24719	1.91E−05	2.01	1.22E−04	2.05
P	230	5.23E−02	—	2.32E−01	—
	944	2.23E−02	1.20	1.36E−01	0.76
	4038	9.32E−03	1.20	6.32E−02	1.05
	16374	4.21E−03	1.14	3.43E−02	0.87

Table 2: Errors and convergence rates for the $\mathbb{P}_3$ scheme with uniform triangular Delaunay primal meshes.

	DOF	E_1	O_1	E_∞	O_∞
U_1	363	3.83E−05	—	1.56E−04	—
	1454	1.98E−06	4.27	1.34E−05	3.54
	6135	1.08E−07	4.04	8.81E−07	3.78
	24719	6.42E−09	4.05	4.58E−08	4.24
U_2	363	3.89E−05	—	1.88E−04	—
	1454	2.04E−06	4.25	1.43E−05	3.71
	6135	1.07E−07	4.09	6.78E−07	4.24
	24719	6.41E−09	4.04	4.75E−08	3.81
P	230	1.24E−03	—	6.86E−03	—
	944	1.34E−04	3.15	1.34E−03	2.32
	4038	1.17E−05	3.36	1.23E−04	3.29
	16374	1.29E−06	3.15	1.82E−05	2.73

Table 3: Errors and convergence rates for the $\mathbb{P}_5$ scheme with uniform triangular Delaunay primal meshes.

	DOF	E_1	O_1	E_∞	O_∞
U_1	363	1.34E−06	—	1.59E−05	—
	1454	1.40E−08	6.57	1.57E−07	6.66
	6135	1.48E−10	6.32	1.01E−09	7.01
	24719	2.66E−12	5.77	1.70E−11	5.87
U_2	363	1.34E−06	—	1.69E−05	—
	1454	1.44E−08	6.53	1.58E−07	6.74
	6135	1.49E−10	6.35	1.31E−09	6.66
	24719	2.76E−12	5.73	1.84E−11	6.12
P	230	4.39E−05	—	4.11E−04	—
	944	6.91E−07	5.88	9.19E−06	5.38
	4038	1.32E−08	5.45	1.12E−07	6.07
	16374	3.65E−10	5.13	4.42E−09	4.62

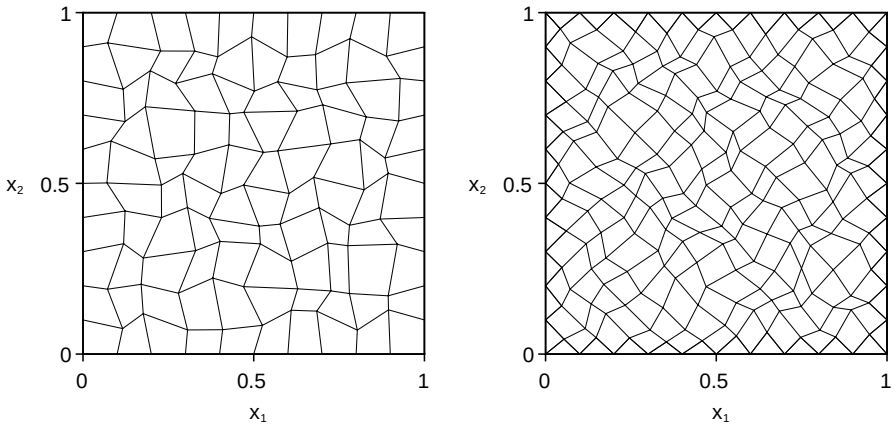

Figure 6: A coarse deformed quadrilateral mesh (left) and the associated diamond mesh (right).

Table 4: Errors and convergence rates for the $\mathbb{P}_1$ scheme with deformed quadrilateral primal meshes.

	DOF	E_1	O_1	E_∞	O_∞
U_1	220	2.06E−03	—	6.78E−03	—
	840	4.58E−04	2.24	1.93E−03	1.88
	3280	1.08E−04	2.12	5.65E−04	1.80
	12960	2.62E−05	2.06	1.48E−04	1.95
U_2	220	1.94E−03	—	8.35E−03	—
	840	4.36E−04	2.23	3.52E−03	1.29
	3280	1.08E−04	2.05	7.74E−04	2.22
	12960	2.63E−05	2.06	2.44E−04	1.68
P	100	6.75E−02	—	2.23E−01	—
	400	2.41E−02	1.48	1.55E−01	0.53
	1600	9.07E−03	1.41	6.83E−02	1.18
	6400	4.02E−03	1.17	3.61E−02	0.93

Table 5: Errors and convergence rates for the $\mathbb{P}_3$ scheme with deformed quadrilateral primal meshes.

	DOF	E_1	O_1	E_∞	O_∞
U_1	220	6.05E−05	—	3.64E−04	—
	840	3.25E−06	4.36	1.98E−05	4.35
	3280	1.83E−07	4.22	1.88E−06	3.46
	12960	1.11E−08	4.08	9.87E−08	4.29
U_2	220	5.65E−05	—	2.97E−04	—
	840	3.06E−06	4.35	2.12E−05	3.94
	3280	1.80E−07	4.16	1.50E−06	3.89
	12960	1.12E−08	4.04	1.25E−07	3.61
P	100	2.10E−03	—	1.03E−02	—
	400	1.89E−04	3.47	1.81E−03	2.51
	1600	1.60E−05	3.56	2.22E−04	3.03
	6400	1.65E−06	3.28	3.09E−05	2.85

Table 6: Errors and convergence rates for the $\mathbb{P}_5$ scheme with deformed quadrilateral primal meshes.

	DOF	E_1	O_1	E_∞	O_∞
U_1	220	3.08E−06	—	2.26E−05	—
	840	3.38E−08	6.73	1.97E−07	7.08
	3280	4.37E−10	6.38	5.28E−09	5.31
	12960	6.35E−12	6.16	5.62E−11	6.61
U_2	220	2.72E−06	—	1.46E−05	—
	840	3.14E−08	6.66	3.20E−07	5.70
	3280	4.43E−10	6.26	3.37E−09	6.68
	12960	6.36E−12	6.18	6.11E−11	5.84
P	100	5.84E−05	—	3.18E−04	—
	400	1.35E−06	5.44	2.12E−05	3.91
	1600	2.92E−08	5.53	4.73E−07	5.48
	6400	8.03E−10	5.18	1.63E−08	4.86

5.2. Assessment of the preconditioning technique

We devote this subsection to assess the quality of the preconditioning technique proposed in Section 4. Two topics are here considered: the use of approximate matrices $\widetilde{A}$, $\widetilde{B}$, and $\widetilde{C}$ deriving from a Patankar-like discretization to compute an approximation of $\mathcal{P}$ and the use of the incomplete inverse matrix for preconditioning. To do so, we consider four situations.

- Accordingly to Section 4, matrix $\mathcal{P}$ defined by relation (11) stands for the ideal preconditioning matrix as shown in [35]. Nevertheless, computation $\mathcal{P}^{-1}$ is not possible since the Schur complement S is not invertible. To overcome such a difficulty, we add a slight perturbation setting $S_\varepsilon = CA^{-1}B^T + \varepsilon I$ with $\varepsilon > 0$ small enough (we take $\varepsilon = 10^{-8}$ in the numerical tests). Since S_ε is now invertible, we define
$$\mathcal{P}_\varepsilon^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0 & S_\varepsilon^{-1} \end{bmatrix}.$$

- To evaluate the impact of the approximate matrices deriving from the Patankar-like discretization, we shall consider a preconditioning matrix $\widetilde{\mathcal{P}}$ based on $\widetilde{A}$, $\widetilde{B}$, and $\widetilde{C}$. One more time, the Schur complement $\widetilde{S} = \widetilde{C}\widetilde{A}^{-1}\widetilde{B}^T$ is not invertible and we slightly perturbe the matrix setting $\widetilde{S}_\varepsilon = \widetilde{S} + \varepsilon I$ such that $\widetilde{\mathcal{P}}_\varepsilon$ is invertible and we set
$$\widetilde{\mathcal{P}}_\varepsilon^{-1} = \begin{bmatrix} \widetilde{A}^{-1} & 0 \\ 0 & \widetilde{S}_\varepsilon^{-1} \end{bmatrix}.$$

- To assess the efficiency of the incomplete inverse procedure, we substitute the inverse operator $\cdot^{-1}$ by the $\cdot^\dagger$ one. To deal with the Schur matrix, we use projector operator $\mathbb{T}_\mathcal{M}$ defined in subsection 4.4 and we set $\widehat{\widetilde{S}} = \mathbb{T}_\mathcal{M}\left(\widetilde{C}\widetilde{A}^\dagger\widetilde{B}^T\right)$. Indeed, matrix $\widetilde{C}\widetilde{A}^\dagger\widetilde{B}^T$ has non-null entries which do not correspond to adjacent cells hence we apply the $\mathbb{T}_\mathcal{M}$ projector to guarantee that matrix $\widehat{\widetilde{S}}$ enjoys the T_0 property given by definition 2. Since $\widehat{\widetilde{S}}$ now satisfies the T_0 property and $\widetilde{A}$ satisfies the T_0 or T_1 properties, we can apply the $\cdot^\dagger$ operator and we define
$$\widetilde{\mathcal{P}}^\dagger = \begin{bmatrix} \widetilde{A}^\dagger & 0 \\ 0 & \widehat{\widetilde{S}}^\dagger \end{bmatrix}.$$

- For the last issue, we want to substitute the inverse operator $\cdot^{-1}$ by the $\cdot^\dagger$ when one employs the exact matrices A, B, and C of the linear system (10). Since S does not enjoy the T_0 property and A does not enjoy the T_0 or T_1 properties for quadrilateral or triangular primal meshes, respectively, a specific treatment is required to apply the $\cdot^\dagger$ operator. For matrix A we simply apply the $\mathbb{T}_\mathcal{D}$ operator setting $\widehat{A} = \mathbb{T}_\mathcal{D}(A)$. To build an approximation of the Schur matrix, we define $\widehat{S} = \mathbb{T}_0\big(C\widehat{A}^\dagger B^T\big)$ which turns to be a matrix of type T_0. We then define the incomplete inverse preconditioning matrix by
$$\mathcal{P}^\dagger = \begin{bmatrix} \widehat{A}^\dagger & 0 \\ 0 & \widehat{S}^\dagger \end{bmatrix}.$$

Remark 6. *We do not explicitly build the matrices A, B, and C since the method only deals with operator $\mathcal{H}$. Nevertheless it is possible to recover the matrix by computing $\mathcal{H}(\Phi)$ where Φ spans the canonical basis. In that way, we obtain all the columns of the matrices. Of course, such a technique is not computationally interesting but it enables to evaluate explicitly all the matrices to carry out the numerical tests of this subsection.*

5.2.1. Efficiency assessments of the preconditioning matrices

To perform the simulations, we consider a uniform triangular Delaunay mesh with 944 primal cells and its associated diamond mesh with 1454 cells corresponding to a linear system with 3852 unknowns. We plot in Fig. 7 the residual curves of the GMRES method for $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, where R^{iter} is the L^2-norm residual at iteration *iter*, with no preconditioning and with preconditioning, where for the latter we consider the four preconditioning matrices just presented. The main observations are the following.

- The exact inverse matrix $\mathcal{P}_\varepsilon^{-1}$ enables to determine the exact solution in 4 iterations as expected and the small perturbation controlled by ε does not perturb the convergence. The preconditioner is efficient for all the reconstructions and suggest that approximation of $\mathcal{P}_\varepsilon^{-1}$ will provide a very good matrix for reducing the conditioning number of matrix $\mathcal{A}$.

- The use of any preconditioning matrix dramatically reduces the number of iterations with respect to the identity matrix (no preconditioning). We also remark that the number of iterations is mainly the same, independent of the polynomial degree reconstruction since the number of unknowns is preserved (only the stencil size changes, hence the matrix connectivity) with the exception of the non-preconditioning case where the computational effort increases with the polynomial degree.

- Substituting the exact matrices A, B, and C with their respective approximations and using the exact inverse matrix $\widetilde{\mathcal{P}}_\varepsilon^{-1}$ provides excellent results, even with higher degree. For the global system with 3832 unknowns we just need about 80 iterations to provide the solution. In other word, the use of Patankar-like matrices to perform the preconditioning of the matrix-free problem is very efficient.

- We now deal with the incomplete inverse preconditioning matrix using the exact matrices or the approximated matrices. We observe that the two incomplete inverse matrices provide approximatively the same iterations numbers which means that the loose of performance may be essentially imputed to the inverse procedure and not the matrix approximation. In all the situations, we cut by 3 or 4 the number of iterations but we recall that the preconditioning procedure is highly parallelizable since it just corresponds to a sparse matrix product with a vector.

As an overall conclusion, the use of the Patankar-like matrices in substitution of the real matrices is very efficient and the incomplete inverse preconditioning enables to substantively reduce the number of iterations of the GMRES procedure.

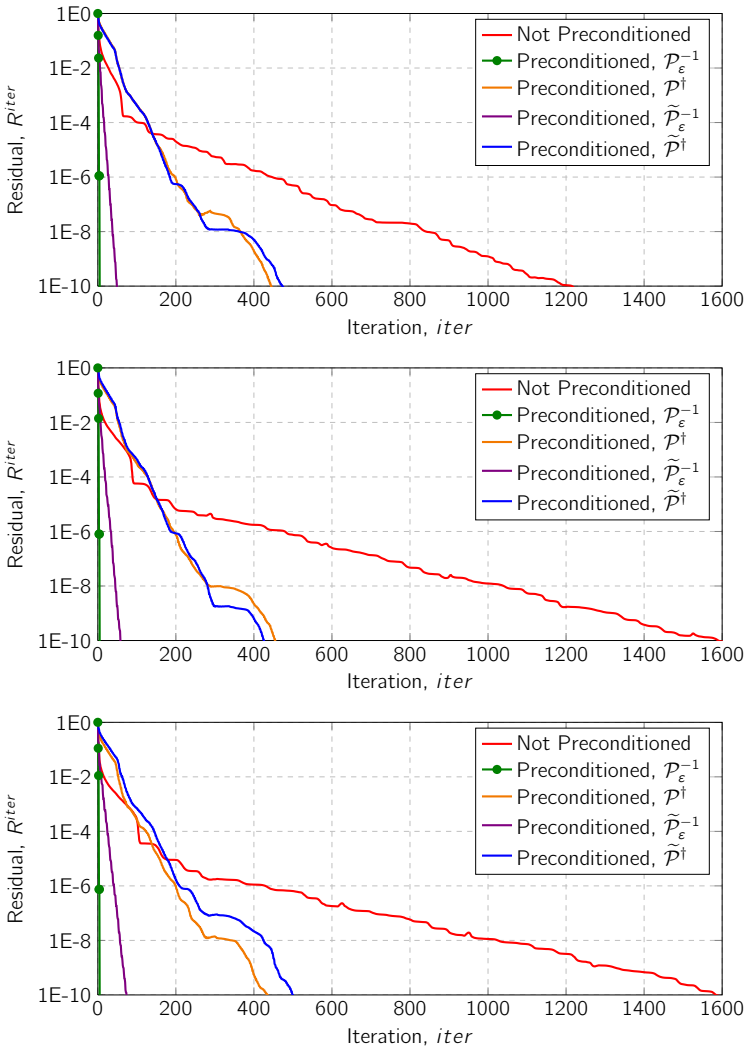

Figure 7: Comparison of the residual curves of the GMRES method with different preconditioning matrices for $\mathbb{P}_1$ (top), $\mathbb{P}_3$ (middle), and $\mathbb{P}_5$(bottom) polynomial reconstructions.

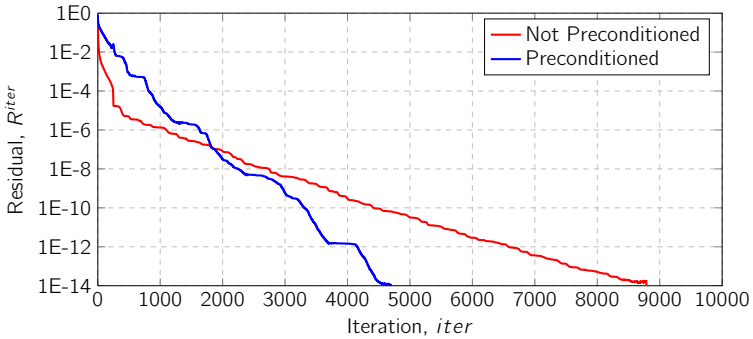

Figure 8: Comparison of the residual curves of the GMRES method with and without preconditioning matrix for $\mathbb{P}_1$ polynomial reconstructions.

5.2.2. Convergence and saturation for fine meshes

We now turn to assess the preconditioning using $\widetilde{\mathcal{P}}^{\dagger}$ when dealing with bigger linear systems. For that purpose, we consider a uniform triangular Delaunay mesh with 16374 primal cells and 24719 diamond cells corresponding to 65812 unknowns.

We plot in Fig. 8 the residual curves of the GMRES method with no preconditioning and with the preconditioning matrix $\widetilde{\mathcal{P}}^{\dagger}$ for the $\mathbb{P}_1$ reconstruction while we present the results for the other reconstruction in Table 7 where N_{ITER} is the number of iterations to achieve a residual of $R^{iter} < 10^{-14}$, T_{E} is the execution time (normalized for the sake of comparison), and T_{PREC} is the percentage of T_{E} in the preconditioning of the residual vectors during the GMRES procedure.

We observe that the non-preconditioning version converges very slowly and the dependance on the polynomial degree is noticeable. On the contrary, the new preconditioning procedure is low sensitive to the polynomial degree. We also note the very low computational cost of the preconditioning procedure, the huge differences of running time between with and without preconditioning, and the increase of the computational effort just by 20% and 60% fot the $\mathbb{P}_3$ and $\mathbb{P}_5$ versions, respectively, which demonstrates the great advantage to perform high-order approximation on coarse meshes when comparing with second-order approximations on fine meshes. Of course fine meshes could be necessary to capture specific structures of the solution (vortex, obstacle, discontinuities) but the very high-order approximation should be privileged when possible.

Table 7: Number of GMRES iterations and execution time using uniform triangular Delaunay meshes.

		N_{ITER}	T_{E}	T_{PREC} [%]
$\mathbb{P}_1$	I_{d}	8792	1.36	—
	$\widetilde{\mathcal{P}}^{\dagger}$	4700	1	3.4
$\mathbb{P}_3$	I_{d}	25000*	3.63*	—
	$\widetilde{\mathcal{P}}^{\dagger}$	4651	1.18	2.9
$\mathbb{P}_5$	I_{d}	43000*	9.15*	—
	$\widetilde{\mathcal{P}}^{\dagger}$	4662	1.61	2.3

* Estimated value

In spite of that positive conclusion, a more representative preconditioning matrix based on the second-order discretization should be considered in the future. Indeed, the matriz approximations $\widetilde{A}$, $\widetilde{B}$, and $\widetilde{C}$ derive from a Patankar-like discretization which is non-consistent for unstructured meshes.

We carried out the same simulation using deformed quadrilateral meshes and we noticed the same effectiveness of the proposed preconditioning as for the presented tests.

5.3. Scalability and time consumption

Multithreading/multicore implementation is of crucial importance when dealing with a large number of unknowns and long running times (larger than several hours for instance). Computers are equipped with multicore processors and GPU cards allowing efficient time simulations cut by running specific portions of the code in parallel. High Performance Computing aims to take advantage of algorithms supporting the multithreading so we have developed and implemented a parallel version of the method using the openMP framework for a small multicore machine to assess the scalability and the potentiality to be parallelized for large multicore machines.

5.3.1. Speed-up assessment

For each simulation, we evaluate the following parameters:

- T_{PREC} — execution time to build the preconditioning matrix;
- T_{GMRES} — execution time to solve the linear system using the GMRES method (including the preconditioning, the Krylov basis construction, and the residual vectors computation);
- T_{E} — execution time of the entire simulation;
- S_n — speedup with n cores given by

$$S_n = \frac{T_{\mathrm{E},1}}{T_{\mathrm{E},n}}, \tag{18}$$

  where $T_{\mathrm{E},1}$ is the execution time with one CPU core and $T_{\mathrm{E},n}$ is the execution time with n CPU cores;

- E_n — strong efficiency with n cores (also given in percentage) is defined as

$$E_n = \frac{S_n}{n}. \tag{19}$$

All the time parameters correspond to the wall-clock time and the measurements are given in seconds. The efficiency of an algorithm is related to the relative usage of each allocated core. Very scalable algorithms have high values of efficiency and the Amdahl's Law predicts the speed-up with n cores as

$$S_n = \frac{1}{1 - P_n + \dfrac{P_n}{n}}. \tag{20}$$

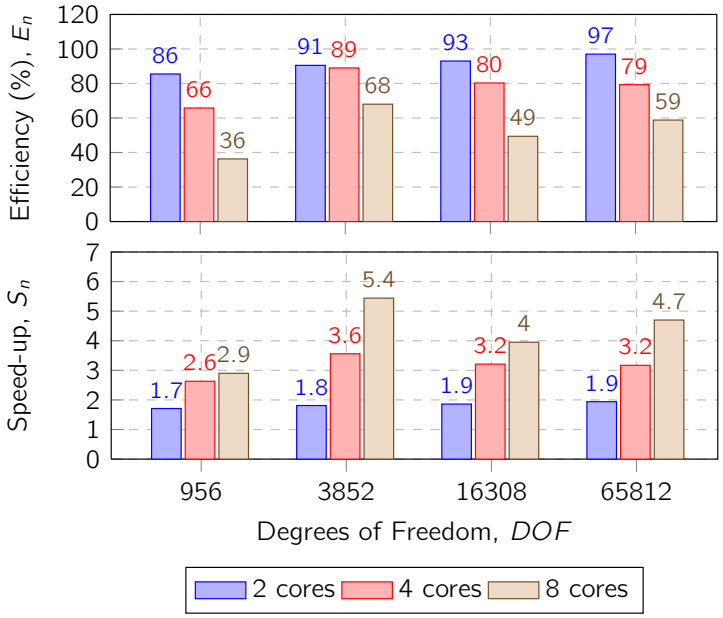

Figure 9: Speed-ups and efficiency using uniform triangular Delaunay meshes with $\mathbb{P}_1$ polynomial reconstructions.

where P_n is the fraction of the algorithm running in parallel. Notice that, according to Amdahl, a linear speed-up is not usually achieved since it implies a full parallel algorihtm, which is not always possible. Moreover, the Amdahl's Law does not take into account some effects such as syncronization, overhead, or cache saturation, which can reduce the speed-up or even increase it. In fact, a superlinear speed-up ($S_n > n$) is, in practice, possible since more CPUs implies a larger available cache memory, avoiding cache saturation effects. Since we do not know, *a priori*, the value of P_n, we cannot predict the speed-up using the Amdahl's Law. Instead, we obtain P_n using equation (20) for a given speed-up and number of cores, as a indicator of the quality of the scalability.

Simulations were carried out with uniform Delaunay meshes and we perform the computation with $n = 1, 2, 4, 8$ cores. Having $T_{\mathrm{E},1}$ and $T_{\mathrm{E},n}$ in hand, we compute the speed-ups with relation (18), the efficiency with equation (19), and finally deduce P_n from the Amdahl's Law (20). The algorithm was implemented in C++ and parallelized using the openMP framework. The machine has 8 Intel Xeon processors with 2.2 GHz of clock rate and 2 MB of cache memory for each core. We display the speed-up and the efficiency for the $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions in Fig.s 9, 10, and 11, respectively, where DOF is the total number of unknowns, *i.e.* $DOF = 2K + I$.

We report a very good scalability using 2 CPU cores with speed-ups close to 2 for the three kinds of reconstruction. We obtain larger speed-ups allocating more CPU cores until reaching a speed-up around 4.5 with 8 cores with small variations with respect to the polynomial degree and the mesh size (except for very small meshes where all the data is contained in the cache memory). Such a situation is expected since more

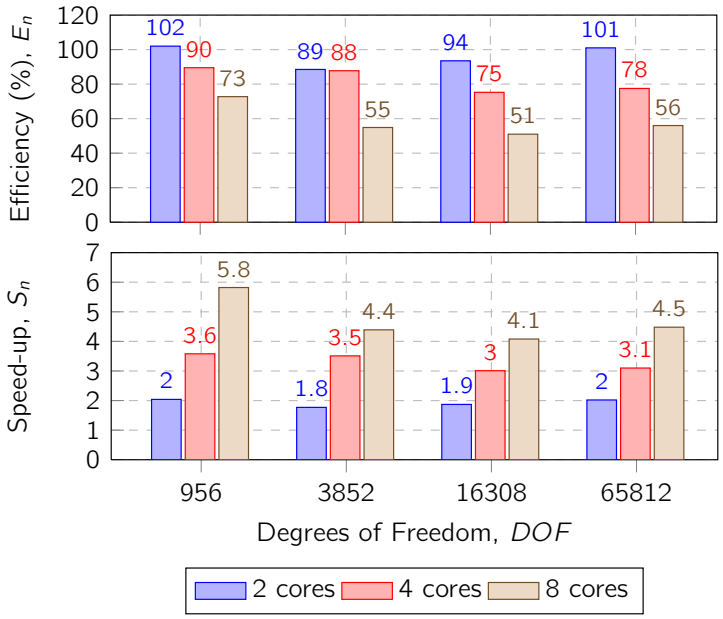

Figure 10: Speed-ups and efficiency using uniform triangular Delaunay meshes with $\mathbb{P}_3$ polynomial reconstructions.

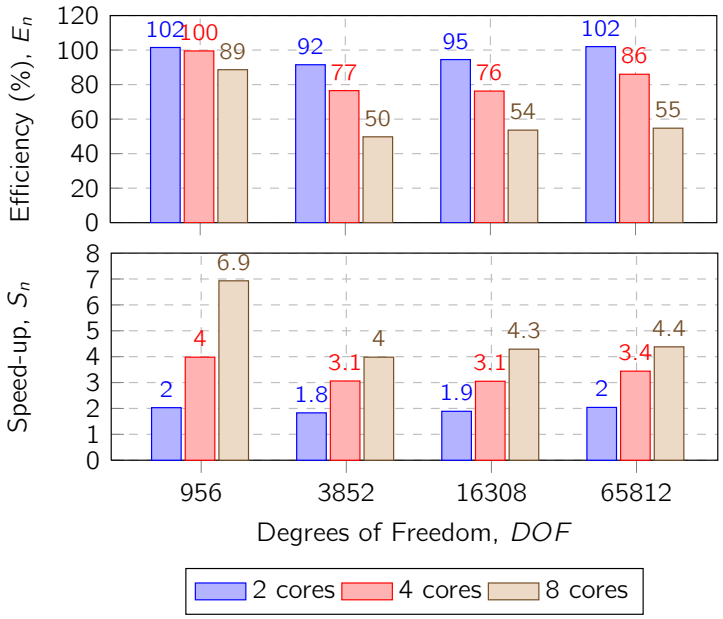

Figure 11: Speed-ups and efficiency using uniform triangular Delaunay meshes with $\mathbb{P}_5$ polynomial reconstructions.

threads lead to larger overhead delays, that is, task start-up time, synchronizations, data communications, task termination time, among other factors, slowing down the simulations. Moreover, the bottleneck between the main and the cache memory slows down the data transfer when dealing with more than two cores. We notice some efficiency rates above 100% because some cache memory effects can lead to a superlinear speed-up in some scenarios. Moreover, the smallest mesh has the worst efficiency (36% with speed-up of 2.9) since the overhead affects very much the simulation.

Considering the Amdahl's Law, we predict a fraction of parallelized code around $P = 90\%$ which is a indicator of a good scalability. Of course, as stated before, the Amdahl's prediction does not take into account some factors which can slow down the algorithm.

To sum up, the tests we have carried out show that the method we propose has a great potential to be parallelized for large multicore machines to drastically reduce the execution time.

5.3.2. Time consumption versus mesh size

Tables 8, 9, and 10 provide the execution time per task and the total execution time, where $DOF = 2K + I$. The time computation of the preconditioning matrices and the incomplete inverse $\widetilde{\mathcal{P}}^{\dagger}$ during the pre-processing stage is only given in Table 8 since it does not depend on the polynomial reconstruction degree.

Table 8: Runtime per task and multithreading performance of the $\mathbb{P}_1$ scheme using uniform triangular Delaunay meshes.

DOF	n	T_{PREC} [s]	T_{GMRES} [s]	T_{E} [s]
956	1	0.04	0.47	0.57
	2	0.02	0.26	0.33
	4	0.01	0.06	0.26
	8	0.01	0.11	0.20
3852	1	0.53	10.10	10.94
	2	0.28	5.44	5.93
	4	0.15	2.97	3.26
	8	0.09	1.84	2.01
16308	1	9.31	238.22	249.65
	2	4.81	128.10	134.17
	4	2.47	81.39	84.58
	8	1.30	61.53	63.28
65812	1	169.73	5319.71	5510.26
	2	87.87	2743.64	2843.42
	4	46.12	1741.87	1794.20
	8	22.57	1145.93	1172.36

Table 9: Runtime per task and multithreading performance of the $\mathbb{P}_3$ scheme using uniform triangular Delaunay meshes.

DOF	n	T_{GMRES} [s]	T_{E} [s]
956	1	0.95	1.39
	2	0.44	0.68
	4	0.26	0.39
	8	0.16	0.24
3852	1	14.23	16.43
	2	8.10	9.28
	4	4.06	4.68
	8	3.39	3.74
16308	1	302.00	319.30
	2	161.46	170.55
	4	101.50	106.19
	8	75.66	78.20
65812	1	6541.88	6759.10
	2	3228.46	3342.11
	4	2123.95	2183.35
	8	1477.90	1508.14

Table 10: Runtime per task and multithreading performance of the $\mathbb{P}_5$ scheme using uniform triangular Delaunay meshes.

DOF	n	T_{GMRES} [s]	T_{E} [s]
956	1	1.95	4.06
	2	0.94	2.00
	4	0.47	1.02
	8	0.28	0.59
3852	1	23.69	32.24
	2	13.11	17.59
	4	8.27	10.53
	8	6.92	8.11
16308	1	477.74	523.93
	2	252.88	276.73
	4	159.29	171.51
	8	115.87	122.14
65812	1	10673.78	11058.73
	2	5234.73	5433.43
	4	3108.89	3215.69
	8	2474.25	2526.57

The pre-processing stage is very low consuming with respect to the whole computational process and the essential cost derives from the GMRES routine. We observe that the $\mathbb{P}_3$ situation generates an overcost of about 30% whatever the number of core and the mesh size are. For the $\mathbb{P}_5$ reconstruction, the situation is less clear: for small mesh size the ratio between the time consumption with $\mathbb{P}_5$ and $\mathbb{P}_1$ ranges between 3 and 4 in function of the cores number. For the two finest meshes, the ratio is stabilized around a factor 2 independently of the core numbers. At last we observe that the time consumption roughly increases as $I^{3/4}$ or, taking into account the space dimension, the time consumption increases as $h^{-3/2}$ where $h = 1/\sqrt{I}$ is the mesh parameter and we observe low sensitivity regarded to the number of cores and the polynomial degree. Of course the time is reduced when employing more cores or lower degree but the time is still a function of order $h^{-3/2}$.

5.3.3. Time consumption versus the error of approximation

We now compare the computational cost of the proposed method in terms of accuracy. For a given tolerance, we want to determine the mesh and the associated execution time one has to use to provide an accurate approximation up to the prescribed error for several type of reconstructions. To do so, we compute the L^2-norm errors of the velocity and the pressure, given by

$$E_2^{\beta}(\mathcal{D}) = \left(\frac{\sum\limits_{k \in \mathcal{C}_{\mathcal{D}}} |c_k| (U^{\star}_{\beta,k} - \overline{U}_{\beta,k})^2}{\sum\limits_{k \in \mathcal{C}_{\mathcal{D}}} |c_k| \overline{U}^2_{\beta}} \right)^{\frac{1}{2}} \quad \text{and} \quad E_2^{P}(\mathcal{M}) = \left(\frac{\sum\limits_{i \in \mathcal{C}_{\mathcal{M}}} |c_i| (P^{\star}_{i} - \overline{P}_{i})^2}{\sum\limits_{i \in \mathcal{C}_{\mathcal{M}}} |c_i| \overline{P}^2_{i}} \right)^{\frac{1}{2}},$$

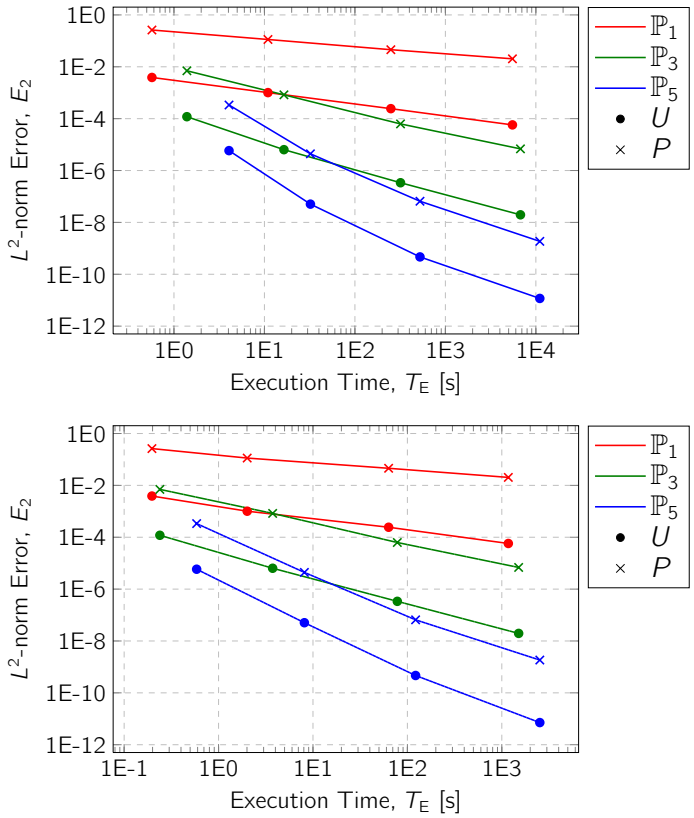

Figure 12: L^2-norm error as a function of the execution time. The simulations were carried out using uniform triangular Delaunay meshes with $DOF = 956, 3852, 16308, 65812$ and one CPU core (top) and eight CPU cores (bottom).

using successive finer uniform triangular Delaunay meshes and the execution time of the simulation, T_E. We plot in Fig. 12 the L^2-norm error as a function of the execution time with $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions for four meshes. Clearly the $\mathbb{P}_5$ reconstruction has the lower computational cost to achieve an approximate solution up to a given tolerance. As an example, to provide a numerical solution with one core such that the L^2-error of the velocity is lower than 10^{-6}, the $\mathbb{P}_5$ version requires 10s with a mesh of 2500 cells, the $\mathbb{P}_3$ reconstruction needs 100s with a mesh of 12000 cells, and the $\mathbb{P}_1$ second-order method performs the computation in 10^8s (about 3 years and 2 months) with a mesh of about 10^7 cells (estimated value).

6. Simulation of a polymer extruder apparatus

We dedicate this section to the polymer extrusion simulation in order to highlight the method capacity to handle complex geometries with unstructured meshes within the very high-order finite volume context. We also intend to evaluate the preconditioning

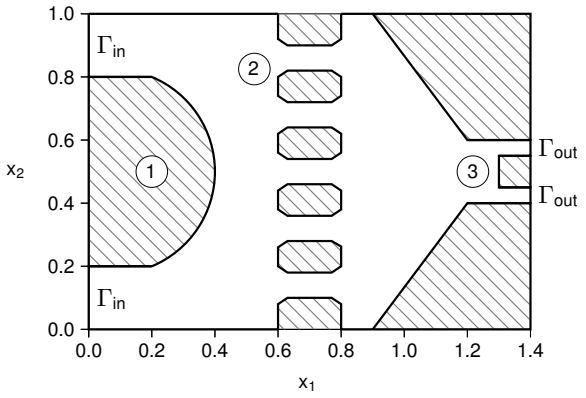

Figure 13: Polymer extruder machine geometry where a rotation screw (1) forces the molten polymer through a die (3) after passing through the breaker plate (2). The breaker plate creates back pressure in the barrel required for uniform melting and proper mixing of the polymer.

technique efficiency in the case of a concrete application. For this purpose, we consider a geometry which corresponds to a standard extruder machine [21] displayed in Fig. 13 where a molten polymer material flows out to the extruder head with a specific constant cross section shape (plastic pipes, window frames, etc.). Due to the high viscosity and the low velocity, the nonlinear contribution of the Navier-Stokes equation is neglected and we assume that the process is governed by the steady-state Stokes equation for incompressible fluids. To prescribe the boundary conditions, we consider an inlet velocity $U_{\mathrm{D,in}}$ defined on $x \in \Gamma_{\mathrm{in}} = \{(x_1, x_2) \in \partial\Omega : x_1 = 0 \wedge (0 < x_2 < 0.2 \vee 0.8 < x_2 < 1)\}$ given by

$$U_{\mathrm{D,in}}(0, x_2) = \begin{cases} (4x_2(0.2 - x_2), 0)\,, & \text{if} \quad 0 < x_2 < 0.2, \\ (6(1 - x_2)(x_2 - 0.8), 0)\,, & \text{if} \quad 0.8 < x_2 < 1, \end{cases}$$

and an outlet velocity $U_{\mathrm{D,out}}$ defined on $x \in \Gamma_{\mathrm{out}} = \{(x_1, x_2) \in \partial\Omega : x_1 = 1.4 \wedge (0.4 < x_2 < 0.45 \vee 0.55 < x_2 < 0.6)\}$ given by

$$U_{\mathrm{D,out}}(1.4, x_2) = \begin{cases} (320(0.45 - x_2)(x_2 - 0.4), 0)\,, & \text{if} \quad 0.4 < x_2 < 0.45, \\ (320(0.6 - x_2)(x_2 - 0.55), 0)\,, & \text{if} \quad 0.55 < x_2 < 0.6. \end{cases}$$

The other portions of the boundary are walls with null velocity $U_{\mathrm{D}}(x) = (0, 0)$. We consider a null source term, that is $f(x) = (0, 0)$, and we assume the viscosity to be unitary, that is $\mu = 1$, for the sake of simplicity since the problem is linear. We carried out three simulations with triangular Delaunay meshes as primal meshes (with 2406, 9354, and 26572 cells) and deduce the respective diamond meshes (with 3750, 14334, and 40380 cells, respectively). We display a coarse version of the primal and dual meshes in Fig. 14. Fig. 15 shows the isocontours of the horizontal (top panel) and vertical (middle panel) components of velocity together with the isocontours of the pressure (bottom panel) for the finer mesh and with $\mathbb{P}_5$ polynomial reconstructions. An enlargement near the extruder outlet is plotted on the right side to detail the flow structures. The isolines are slightly asymmetric since the inflow conditions are not the same in the upper and lower

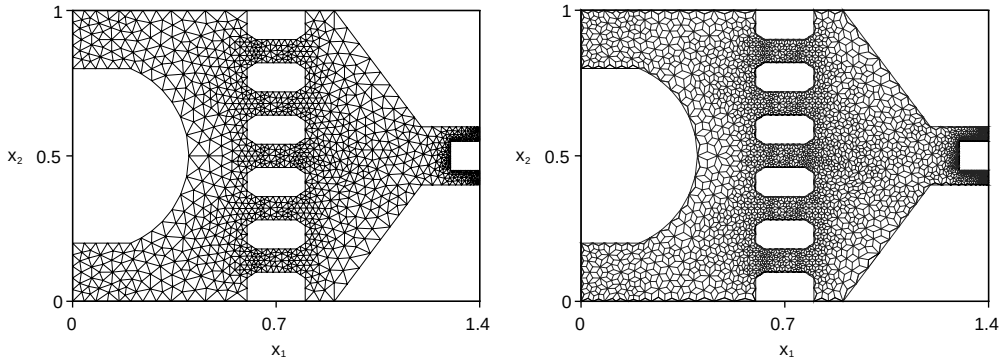

Figure 14: A coarse non-uniform triangular Delaunay mesh (left) and its associated diamond mesh (right).

inflow pipes. To assess the accuracy of the different polynomial reconstruction, we report in Table 11 the minimum and the maximum mean values of the three scalar quantities, where $DOF = K$ for U_1 and U_2 and $DOF = I$ for P. The minimum of the horizontal and vertical velocities presents the main variations. We notice that the fourth- and sixth-order schemes provide very similar values whereas we observe larger differences between the second- and fourth-order ones. The most noteworthy difference occurs with the pressure where the $\mathbb{P}_5$ reconstruction with the smaller mesh provides the same approximation than the $\mathbb{P}_1$ reconstruction with the middle size mesh. The fourth- and sixth-order methods provide very similar pressure results with the finer mesh (difference of about 7 units for the maximum) whereas the second-order scheme is 80 units far from the maximum pressure using the $\mathbb{P}_5$ reconstruction.

Table 11: Maximum and minimum mean values for U_1, U_2, and P.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	Min	Max	Min	Max	Min	Max
	3750	−1.17E−3	1.98E−1	−1.64E−3	1.99E−1	−2.35E−3	2.00E−1
U_1	14334	−6.37E−4	2.00E−1	−2.31E−4	2.00E−1	−2.53E−4	2.00E−1
	40380	−7.36E−4	2.00E−1	−1.10E−4	2.00E−1	−1.83E−4	2.00E−1
	3750	−7.05E−2	7.45E−2	−7.85E−2	7.51E−2	−7.65E−2	7.43E−2
U_2	14334	−7.59E−2	7.63E−2	−7.74E−2	7.73E−2	−7.75E−2	7.71E−2
	40380	−7.78E−2	7.76E−2	−7.82E−2	7.82E−2	−7.81E−2	7.81E−2
	2406	−80948.13	9868.84	−88824.25	10707.79	−88096.95	10711.55
P	9354	−88784.94	10699.58	−90888.84	10938.38	−90796.49	10932.28
	26572	−90753.83	10907.28	−91526.43	10993.90	−91472.65	10987.23

To assess the efficiency of the preconditioning method, we have carried out the previous simulations with and without the inverse preconditioning matrix and we plot the L^2-norm residuals curves for the $\mathbb{P}_1$ polynomial reconstruction in Fig. 16 (we do not represent the curves with the other reconstructions since they are very similar). We report

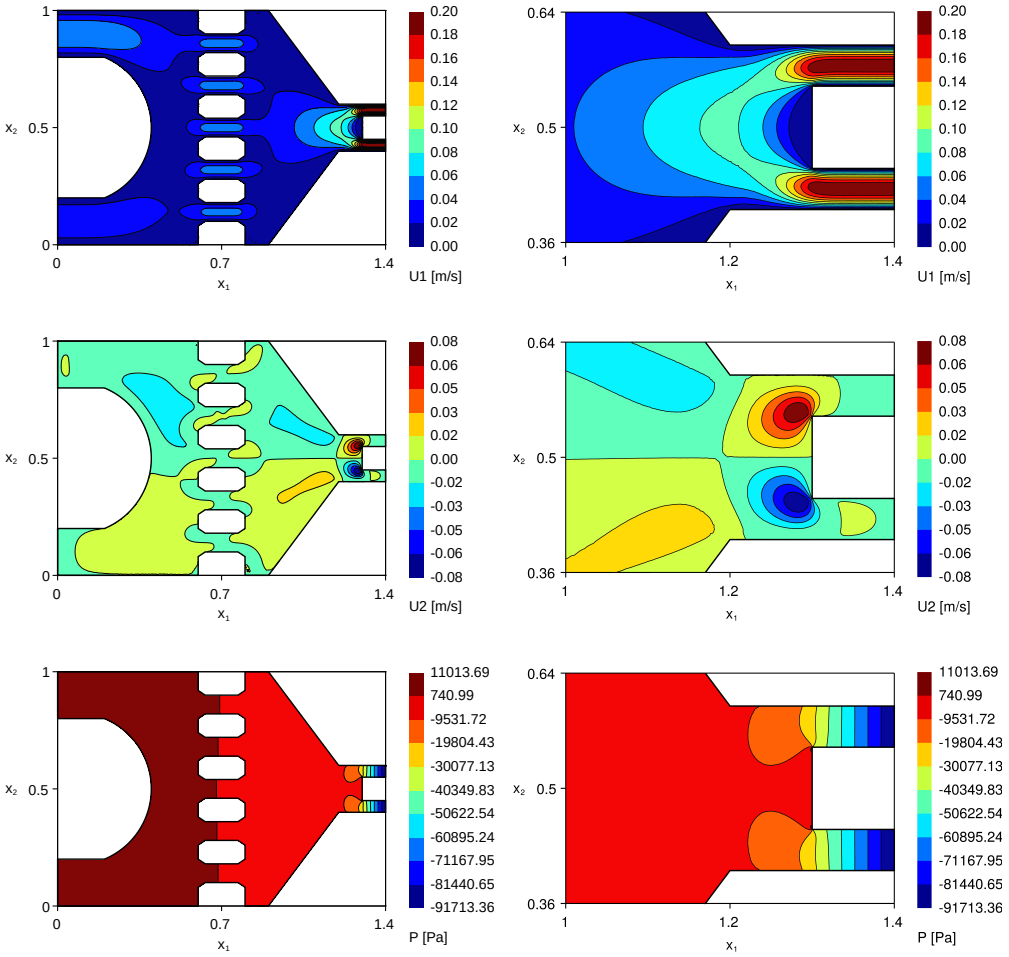

Figure 15: Isocontours of the x_1-component of the velocity (top), isocontours of the x_2-component of the velocity (middle), and isocontours of the pressure (bottom), and respective exit zooms.

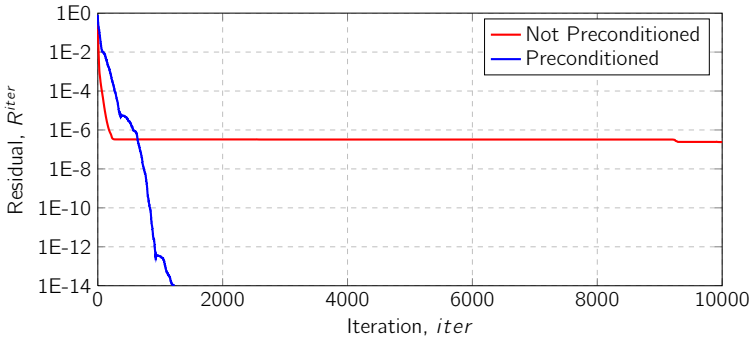

Figure 16: Residual curves of the GMRES method with and without preconditioning matrix and $\mathbb{P}_1$ polynomial reconstructions for a mesh corresponding to 9906 unknowns.

Table 12: Number of GMRES iterations (N_{ITER}), final L^2-norm residual (R_{ITER}), and execution time in seconds (T_{E}).

DOF		9906			38022			107332		
		N_{ITER}	R_{ITER}	T_{E} [s]	N_{ITER}	R_{ITER}	T_{E} [s]	N_{ITER}	R_{ITER}	T_{E} [s]
$\mathbb{P}_1$	I_{d}	NC	2.39E−07	—	NC	8.06E−08	—	NC	1.19E−07	—
	$\widetilde{\mathcal{P}}^{\dagger}$	1220	9.83E−15	10.30	2805	9.86E−15	161.55	5184	9.97E−15	1621.02
$\mathbb{P}_3$	I_{d}	NC	1.15E−07	—	NC	3.23E−08	—	NC	1.41E−08	—
	$\widetilde{\mathcal{P}}^{\dagger}$	1417	9.38E−15	17.01	2555	9.95E−15	191.67	5043	9.93E−15	1691.06
$\mathbb{P}_5$	I_{d}	NC	7.18E−08	—	NC	2.02E−08	—	NC	9.15E−09	—
	$\widetilde{\mathcal{P}}^{\dagger}$	1599	9.59E−07	29.49	3109	9.71E−15	299.33	6513	1.00E−14	2180.70

NC: No Convergence

in Table 12 a set of output parameters for the simulations we have done.

We observe that the non-preconditioning versions do not converge and the residual norm saturate above the prescribe tolerance threshold. We report the minimum residual we achieved and notice the low sensitivity to the polynomial degree and the high dependency to the mesh size. When the preconditioning technique is applied, the GMRES method always converge up to the prescribed tolerance and we report the number of iterations that took to obtain convergence and the associated running time.

7. Conclusion

We have presented an high-order finite volume scheme to solve the bidimensional incompressible Stokes problem based on a new class of polynomial reconstructions. The scheme achieves an effective sixth-order accuracy for the velocity and the fifth-order accuracy for the pressure. We have also presented a new preconditioning technique based on the Schur complement. A large number of numerical tests were carried out to prove its high-efficiency in reducing the computational effort.

Acknowledgements

This research was financed by FEDER Funds through Programa Operational Fatores de Competitividade — COMPETE and by Portuguese Funds FCT — Fundação para a Ciência e a Tecnologia, within the Projects PEst-C/MAT/UI0013/2014 and FCT-ANR/MAT-NAN/0122/2012.

References

[1] T.J. Barth, P.O. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper 90-0013, 1990.
[2] T.J. Barth, Recent developments in high order k-exact reconstruction on unstructured meshes, AIAA Paper 93-0668, 1993.
[3] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numerica (2005) 1–137.
[4] B.J. Boersma, A 6th order staggered compact finite difference method for the incompressible Navier-Stokes and scalar transport equations, J. Comput. Phys. 230 (2011) 4940–4954.
[5] S. Boivin, F. Cayré, J.-M. Hérard, A finite volume method to solve the Navier-Stokes equations for incompressible flows on unstructured meshes, Int. J. Therm. Sci. 39 (2000) 806-825.
[6] A. Boularas, S. Clain, F. Baudoin, A sixth-order finite volume method for diffusion problem with curved boundaries, HAL preprint, https://hal.archives-ouvertes.fr/hal-01052517.
[7] F. Brezzi, M. Fortin, Mixed and hybrid finite element methods, Springer, Berlin-Heidelberg-New York, 1991.
[8] A.J. Chorin, Numerical method for solving incompressible viscous flow problems J. Comp. Phys. 212 (1967) 12-26.
[9] S. Clain, G.J. Machado, J.M. Nóbrega, R.M.S. Pereira, A sixth-order finite volume method for the convection-diffusion problem with discontinuous coefficients, Computer Methods in Applied Mechanics and Engineering 267 (2013) 43–64.
[10] S. Delcourte, D. Jennequin, Preconditioning NavierStokes problem discretized by discrete duality finite volume schemes, in: R. Eymard and J.M. H érard (Eds.), Proceedings of the 5th International Symposium on Finite Volumes for Complex Applications V (2008) 329–336.
[11] S. Delcourte, D. Jennequin, Saddle point preconditioners for linearized Navier-Stokes equations discretized by a finite volume method, Appl. Numer. Math. 60 (2010) 1054–1066
[12] S. Diot, R. Loubère, S. Clain, The MOOD method in the three-dimensional case: very-high-order finite volume method for hyperbolic systems, Int. J. Numer. Meth. Fl. 73 (2013) 362–392.
[13] A. Ern, J.-L. Guermond, Theory and Practice of Finite Elements, vol. 159, Springer Verlag, New-York, 2004.
[14] R. Eymard, T. Gallouët, R. Herbin, Finite volume methods, Handbook of numerical analysis, Vol. VII, pp. 713–1020, North-Holland, Amsterdam, 2000.
[15] R. Eymard, J.C. Latché, R. Herbin, B. Piar, Convergence of a locally stabilized collocated finite volume scheme for incompressible flows, M2AN 43 (2009) 889–927.
[16] E. Ferrer, R.H.J. Willden, A high order discontinuous Galerkin finite element solver for the incompressible Navier-Stokes equations, Comput. & Fluids 46 (2011) 224–230.
[17] J.H. Ferziger, M. Perić, Computational methods for fluids dynamics, Springer-Verlag, Berlin, 1996.
[18] A. Fosso, H. Deniau, F. Sicot, P. Saguat, Curvilinear finite volume schemes unising high-order compact interpolation, J. Comput. Phys., 229 (2010) 5090–5122.
[19] J. Frochte, W. Heinrichs, A splitting technique of higher order for the Navier-Stokes equations, J. Comput. and App. Math. 228 (2009) 373–390.
[20] W. Gao, Y.-L. Duan, R.-X. Liu, The finite volume projection method with hybrid unstructured triangular collocated grids for incompressible flows, J. Hydrodyn. 21 (2009) 201–211.
[21] N.D. Gonçalves, O.S. Carneiro, J.M. Nóbrega, Design of complex profile extrusion dies through numerical modeling, J. Non-Newton. Fluid Mech. 200 (2013) 103–110.
[22] B.E. Griffith, An accurate and efficient method for the incompressible Navier-Stokes equations using the projection method as preconditioner, J. Comput. Phys. 228 (2009) 7565–7595.
[23] J.L. Guermont, P. Minev, J. Shen, An overview of the projection methods for incompressible flows, Comput. Meth.Appl. Engrg. 195 (2006) 6011–6045.

[24] J. Haslinger, T. Kozubek, R. Kučera, G. Peichl, Projected Schur complement method for solving non-symmetric systems arising from a smooth fictitious domain approach, Numer. Linear Algebra Appl. 14 (2007) 713–739.
[25] R.-S. Hirsh, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, J. Comput. Phys. 19 (1975) 90–109.
[26] A. Hokpunna, M. Manhart, Compact fourth-order finite volume method for numerical solutions of Navier-Stokes equations on staggered grids, J. Comput. Phys. 229 (2010) 7545–7570.
[27] B.R. Hutchinson, G.D. Raithby, A multigrid method based on the additive correction strategy, Numerical Heat Transfer 9 (1986) 511–537.
[28] L. Ivan, C.P.T. Groth, High-order solution-adaptative central essentially non-oscillatory (CENO) method for viscous flows, AIAA Paper 2011-367, 2011.
[29] N.A. Kampanis, J.A. Ekaterinaris, A staggerid grid, high-order accurate method for the incompressible Navier-Stokes Equations, J. Comput. Phys. 215 (2006) 589–613.
[30] C.M. Klaij, C. Vuik, SIMPLE-type preconditioners for cell-centered, colocated finite volume discretization of incompressible Reynolds-averaged Navier-Stokes equations, Int. J. Numer. Meth. Fl. 17 (2013) 830–849.
[31] C. Lacor, S. Smirnov, M. Baelmans, A finite volume formulation of compact central schemes on arbitrary structural grids, J. Comput. Phys. 198 (2004) 535–566.
[32] R.D. Lonsdale, An algebraic multigrid scheme for solving the NavierStokes equations on unstructured meshes, in: C. Taylor, J.H. Chin, G.M. Homsy (Eds.), Numerical Methods in Laminar and Turbulent Flow 7 (2), Pineridge Press, Swansea, UK, 1991, pp. 1432–1442.
[33] C. Michalak, C. Ollivier-Gooch, Unstructured high-order accurate finite-volume solutions of the Navier-Stokes equations, AIAA Paper 2009-954, 2009.
[34] A. Montlaur, S. Fernandez-Mendez, A. Huerta, Discontinuous Galerkin methods forthe Stokes equations using divergence-free approximations, Inter. J. Numer. Meth. Fluids 57 (2008) 1071–1092.
[35] M.F. Murphy, G.H. Golub, A.J. Wathen, A Note on preconditioning for indefinite linear systems, SIAM J. Sci. Comput. 21 (2000) 1969–1972.
[36] A. Nigro, C. De Bartolo, F. Bassi, A. Ghidoni, Up to sixth-order accurate A-stable implicit schemes applied to the discontinuous Galerkin discretized Navier-Stokes equations, J. Comput. Phys. 276 (2014) 136–162.
[37] X. Nogueira, S. Khelladi, I. Colominas, L. Cueto-Felgueroso, J. París, H. Gómez, High Resolution finite-volume methods on unstructured grids for turbulence and aeroacustics, Arch. Comput. Meth. E. 18 (2011) 315–340.
[38] C. Ollivier-Gooch, M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation, J. Comput. Phys. Arch. 181(2) (2002) 729-752.
[39] S.V. Patankar, Numerical heat transfer and fluid flow, Hemisphere, New-York, 1980.
[40] J.M.C. Pereira, M.H. Kobayashi, J.C.F. Pereira, A fourth-order-accurate Finite volume Ccmpact method for the incompressible Navier-Stokes solutions, J. Comput. Phys. 167 (2001) 217–243.
[41] R. Peyret, T.D. Taylor, Computational methods for fluid flow, Springer-Verlag, New York, 1985.
[42] M. Piller, E. Stalio, Finite volume compact schemes on staggered grids, J. Comput. Phys. 197 (2004) 299–340.
[43] O. Pironneau, Finite element methods for fluids, John Wiley, Chichester, 1983.
[44] L. Ramírez, X. Nogueira, S. Hhelladi, J. Chassaing, I. Colominas, A new higher-order finite volume method based on moving least squares for the resolution of the incompressible Navier-Stokes equations on unstructured grids, Comput. Meth. Appl. Mech. Engrg. 278 (2014) 883–901.
[45] M. Rehman, C. Vuik, G. Segal, Preconditioners for the steady incompressible Navier-Stokes problem, IAENG Int. J. Appl. Math. 38 (4) (2008) 223–232.
[46] C.M. Rhie, W.L. Chow, A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532.
[47] Y. Sadd, M.H. Schultz, GMRES: a general minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.
[48] Y. Saad, Iterative methods for spares linear systems, Society for Industrial and Applied Mathematics, 2003.
[49] S. Shang, X. Zhao, S. Bayyuk, Generalized formulations for the Rhie-Chow interpolation, J. Comput. Phys. 258 (2014) 880–914.
[50] D. Silvester, H. Elman, D. Kay, A. Wathen, Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow, J. Comput. Appl. Math. 128 (2001) 261–279
[51] S. Smirnov, C. Lacor, M. Baelmans, A finite volume formulation for compact scheme with application to LES, AIAA Paper 2001-2546, 2001.

[52] R. Temam, Navier-Stokes equations. Theory and numerical analysis, North-Holland, Amsterdam, 1987.
[53] D. Vidović, A. Segal, P. Wesseling, A superlinearly convergent finite volume method for the incompressible Navier-Stokes equations on staggered unstructured grids, J. Comput. Phys. 198 (2004) 159–177.
[54] O.C. Zienkiewicz, R.L. Taylor, P. Nithiarasu, The finite element method for fluid dynamics, Butterworth-Heinemann, Waltham, 2014.