A Very High-Order Accurate Staggered Finite Volume Scheme for the Stationary Incompressible Navier–Stokes and Euler Equations on Unstructured Meshes

Ricardo Costa, Stéphane Clain, Gaspar Machado, Raphaël Loubère

Journal of Scientific Computing manuscript No.
(will be inserted by the editor)

A sixth-order accurate staggered finite volume scheme for the incompressible Navier-Stokes and Euler equations on unstructured meshes

Ricardo Costa · Stéphane Clain · Gaspar J. Machado · Raphaël Loubère

Abstract We propose a sixth-order staggered finite volume scheme based on polynomial reconstructions to achieve high accurate numerical solutions for the incompressible Navier-Stokes and Euler equations. The scheme is equipped with a fixed-point algorithm with solution relaxation to speed-up the convergence and reduce the computation time. Numerical tests are provided to assess the effectiveness of the method to achieve up to sixth-order con-

R. Costa
Institut de Mathématiques de Toulouse, Université Paul Sabatier, 31062 Toulouse, France and
Centro de Matemática, Universidade do Minho, *Campus* de Azurém, 4800-058 Guimarães, Portugal
Tel.: +351-253-510400
Fax: +351-253-510401
E-mail: rcosta@math.univ-toulouse.fr

S. Clain
Centro de Matemática, Universidade do Minho, *Campus* de Azurém, 4800-058 Guimarães, Portugal and
Institut de Mathématiques de Toulouse, Université Paul Sabatier, 31062 Toulouse, France
Tel.: +351-253-510400
Fax: +351-253-510401
E-mail: clain@math.uminho.pt

G.J. Machado
Centro de Matemática, Universidade do Minho, *Campus* de Azurém, 4800-058 Guimarães, Portugal
Tel.: +351-253-510400
Fax: +351-253-510401
E-mail: gjm@math.uminho.pt

R. Loubère
Institut de Mathématiques de Toulouse, Université Paul Sabatier, 31062 Toulouse, France
Tel.: +33-561557652
Fax: +33-561558385
E-mail: raphael.loubere@math.univ-toulouse.fr

vergence rates. Simulations for the benchmark lid-driven cavity problem are also provided to highlight the benefit of the proposed high-order scheme.

Keywords Finite volume method · High-order scheme · Polynomial reconstruction · Navier-Stokes equations · Euler equations · Fixed-point algorithm

1 Introduction

Very high-order accurate schemes, that is, schemes where is achieved a strictly higher order than the usual second-order of approximation, for incompressible fluid flow have been first designed using the compact finite difference method on structured grids [19, 1, 26, 23]. On the other hand, following the pioneer work of Patankar [30], the finite volume method turns out to be an efficient way to solve the Navier-Stokes equations for incompressible flow. Nevertheless, most of the finite volume schemes proposed in the literature [5, 15, 7, 17, 22, 38] are based on affine reconstructions of the solution which provides, at most, an effective second-order approximation for the velocity and a first-order approximation for the pressure, either on structured or unstructured meshes. More recently, a new class of very-high order schemes combining a compact scheme and a finite volume method were developed for the Navier-Stokes equations [24, 31, 4, 20, 28] using staggered grids [32, 37, 22] or the collocated method, where the pressure-velocity coupling is enforced by the Rhie-Chow strategy [34, 35, 14].

Up to our knowledge, there exist very few pure finite volume schemes which achieve a very high-order approximation for the incompressible Stokes and Navier-Stokes equations. We mean by "pure", the fact that the Padé technique is not used to provide accurate approximations (pointwise or mean-values) of the unknowns functions and their derivatives, but only polynomial reconstructions are used. The pure version easily handles unstructured

meshes while the compact method turns out to be cumbersome when dealing with unstructured grids [28,16] and a loss of the optimal order of accuracy is also reported. Recently, pure high-order finite volume methods have been proposed in [10,33,27], based on the Moving Least Square method [25] for obtaining the local reconstructions. This approach enables to achieve up to an effective fourth-order of approximation and handles complex meshes. We also mention the contribution [29,21] for compressible flows but such an approach does not fit with the incompressible situation since the duality div-grad does not exist in that case.

We propose in the present paper another pure finite volume scheme, based on specific polynomial reconstructions, to provide sixth-order approximations for the incompressible Navier-Stokes equations and the Euler system. In [11], a new class of polynomial reconstructions have been designed to provide very accurate approximations of the solution for the convection-diffusion problem. An extension for the three-dimensional case with curved boundaries [6] has been developed while the non-stationary problem with time-dependent Dirichlet or Neumann conditions has been studied in [8]. The main ingredient is a differentiated polynomial reconstruction depending on the operator we are dealing with. For the convective part, conservative reconstruction deriving from the one used in hyperbolic problems [12] is considered. For the diffusion term, the non-conservative reconstruction associated to the interface between two cells is desirable to avoid the odd-even solution decoupling problem. At last, the treatment of the Dirichlet condition on polygonal or curved boundary requires a specific form of the conservative reconstruction associated to the edge or face (for 2D and 3D problems, respectively) on the boundary. More recently, the polynomial reconstruction machinery was adapted for the Stokes problem [9] for staggered unstructured grids using the primal mesh for the pressure and the diamond mesh for the velocity. The new polynomial reconstructions which have been proposed provide a sixth-order of approximation while no checkerboard pressure solutions are reported. In this work, a coupled approach is

adopted where the pressure and the velocity are treated as a single vectorial unknown (as such no projection or segregated methods are needed).

The objective of the present paper is twofold. On the one hand, we aim to extend the resolution of the Stokes problem to the nonlinear Navier-Stokes equations and, on the other hand, to provide a robust enough method to tackle the incompressible Euler case, namely taking the viscosity equals to zero. As in the convection-diffusion case, different polynomial reconstructions have to be designed in order to specifically address the operators where they are used. Roughly speaking, conservative reconstruction is dedicated for the convection operator and the pressure whereas the viscous term is discretized with non-conservative polynomial functions. The structure of the paper is the following. The second section deals with the linearization of the Navier-Stokes problem and presents the generic finite volume method. We detail all the specific reconstructions we use to provide a very accurate discretization in section three while section four is dedicated to the fixed-point algorithm. The fifth section presents the numerical simulations we have carried out to prove the effectiveness of the method. Accuracy and robustness as well as solving the incompressible Euler problem are the main issues we address in this section. The lid-driven square cavity problem is considered in section six and the paper ends with a conclusion and some perspectives.

2 Formulation and finite volume discretization

Let Ω be an open bounded polygonal domain of $\mathbb{R}^2$ with boundary $\partial\Omega$ and the outward unit normal vector $n \equiv n(x)$ at $x = (x_1, x_2)$. We seek the velocity field function $U = (U_1, U_2) \equiv (U_1(x), U_2(x))$ and the pressure function $P \equiv P(x)$, which correspond to the solution of the isothermal steady-state flow of an incompressible Newtonian fluid governed by the steady-

state Navier-Stokes equations

$$\nabla \cdot \left(U \otimes U - \nu \nabla U + \frac{1}{\rho} PI \right) = g, \qquad \text{in } \Omega, \tag{1}$$

and the incompressibility constraint given by

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

where constants ν and ρ stand for the kinematic viscosity and the density, respectively, function $g = (g_1, g_2) \equiv (g_1(x), g_2(x))$ stands for the body accelerations per unit volume acting on the continuum, and I stands for the identity matrix in $\mathbb{R}^{2\times 2}$. Tensor ∇U is defined as $[\nabla U]_{\alpha\beta} = \partial U_\alpha / \partial x_\beta$, $\alpha, \beta = 1, 2$, while operator $\otimes$ stands for the tensor product, namely $[U \otimes U]_{\alpha\beta} = U_\alpha U_\beta$. Notice that we do not introduce the non-dimensional system at this stage since we shall also deal with the Euler system *i.e.* setting $\nu = 0$. The system of equations (1-2) is completed with the Dirichlet boundary condition

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

where $U_{\mathrm{D}} = (U_{1,\mathrm{D}}, U_{2,\mathrm{D}}) \equiv (U_{1,\mathrm{D}}(x), U_{2,\mathrm{D}}(x))$ satisfies the divergence-free compatibility condition

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

All the functions are assumed to be regular (at least C^2) in the appropriate domain.

2.1 Linearization

In order to numerically solve the Navier-Stokes problem, a linearized formulation is designed. Let us consider the field function $\Lambda = (\Lambda_1, \Lambda_2) \equiv (\Lambda_1(x), \Lambda_2(x))$, which stands for a guess of the velocity. We consider now the linearized system (also known as the Oseen equation)

$$\nabla \cdot \left(U \otimes \Lambda - \nu \nabla U + \frac{1}{\rho} PI \right) = g, \qquad \text{in } \Omega, \tag{4}$$

$$\nabla \cdot U = 0, \qquad \text{in } \Omega, \tag{5}$$

with the Dirichlet boundary condition (3).

2.2 Primal and diamond meshes

To deal with staggered grids, we introduce the primal and diamond meshes used for the pressure and the velocity, respectively (see Fig. 1).

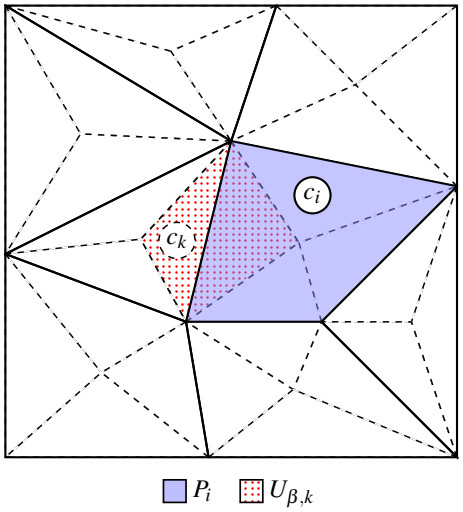

Fig. 1 Primal cells (solid lines) and diamond cells (dashed lines).

The primal mesh of Ω, that we denote by $\mathscr{M}$, is a partition of Ω into I non-overlapping without gap convex polygonal cells c_i, $i \in \mathscr{C}_{\mathscr{M}} = \{1,\dots,I\}$. We adopt the following conventions (see Fig. 2, left):

- for any cell c_i, $i \in \mathscr{C}_{\mathscr{M}}$, 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, $i \in \mathscr{C}_{\mathscr{M}}$, we associate the index set $\nu(i) \subset \{1,\dots,I\} \cup \{\mathrm{D}\}$ 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 $j = \mathrm{D}$.

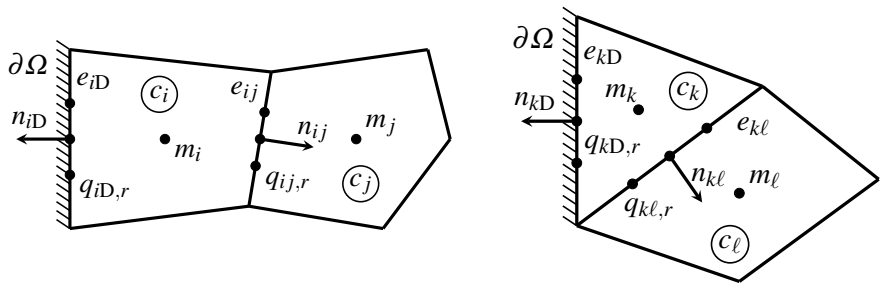

Fig. 2 Notation for the primal mesh (left) and for the diamond mesh (right).

The diamond mesh $\mathscr{D}$ derives from the primal mesh $\mathscr{M}$ and is constituted of K non-overlapping diamond-shape cells (which degenerate to triangles in the boundary) c_k, $k \in \mathscr{C}_{\mathscr{D}} = \{I+1,\dots,I+K\}$. 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 diamond cell associated to the 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 $i \in \mathscr{C}_{\mathscr{M}}$ with $k \in \mathscr{C}_{\mathscr{D}}$ and the index $j \in \nu(i)$ with $\ell \in \nu(k)$ (see Fig. 2, right). In particular m_k is any point in c_k and $m_{k\ell}$ is any point in $e_{k\ell}$. Likewise, for the diamond mesh those are the centroid and the midpoint, respectively.

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

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

2.3 Generic finite volume scheme

Integrating equation (4) over each diamond cell c_k, $k \in \mathscr{C}_{\mathscr{D}}$, and then applying the divergence theorem, provide the conservative law over the cell

$$\int_{\partial c_k} \left(U \otimes \Lambda - \nu \nabla U + \frac{1}{\rho} P I \right) \cdot n \, \mathrm{d}s = \int_{c_k} g \, \mathrm{d}x.$$

We rewrite the expression under the scalar form as

$$\int_{\partial c_k} \left(U_\beta (\Lambda \cdot n) - \nu \nabla U_\beta \cdot n + \frac{1}{\rho} P n_\beta \right) \mathrm{d}s = \int_{c_k} g_\beta \, \mathrm{d}x, \qquad \beta = 1,2.$$

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

$$\sum_{\ell\in\nu(k)} |e_{k\ell}| \left[\sum_{r=1}^{R} \zeta_r (\mathbb{F}^{\otimes}_{\beta,k\ell,r} + \mathbb{F}^{U}_{\beta,k\ell,r} + \mathbb{F}^{P}_{\beta,k\ell,r}) \right] - |c_k| g_{\beta,k} = \mathscr{O}(h_k^{2R}), \qquad \beta = 1,2, \tag{6}$$

with the physical fluxes given by

$$\begin{aligned} &\mathbb{F}^{\otimes}_{\beta,k\ell,r} = U_\beta(q_{k\ell,r})(\Lambda(q_{k\ell,r}) \cdot n_{k\ell}), \qquad \mathbb{F}^{U}_{\beta,k\ell,r} = -\nu \nabla U_\beta(q_{k\ell,r}) \cdot n_{k\ell} \\ &\mathbb{F}^{P}_{\beta,k\ell,r} = \frac{1}{\rho} P(q_{k\ell,r}) n_{\beta,k\ell}, \end{aligned} \tag{7}$$

with $h_k = \max_{\ell\in\nu(k)} |e_{k\ell}|$, and $g_{\beta,k}$ an approximation of order $2R$ of the mean value of g_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].

In the same way, we integrate equation (5) over each primal cell c_i, $i \in \mathscr{C}_{\mathscr{M}}$, and applying again the divergence theorem, we get

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

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

$$\sum_{j\in\nu(i)} |e_{ij}| \sum_{r=1}^{R} \zeta_r \mathbb{F}^{\nabla}_{ij,r} = O(h_i^{2R}), \tag{8}$$

with and $h_i = \max_{j\in\nu(i)} |e_{ij}|$ and the physical flux given by

$$\mathbb{F}^{\nabla}_{ij,r} = U(q_{ij,r}) \cdot n_{ij}. \tag{9}$$

3 Polynomial reconstructions and high-order scheme

Polynomial reconstruction, initially introduced in [2,3] for hyperbolic problems, is the fundamental tool to provide accurate local representations of the underlying solution and provide high-order approximation of the flux in the finite volume context. In [11] a methodology was proposed for the scalar convection-diffusion equation where very accurate approximations of the gradient fluxes is achieved. The method is based on different types of polynomial reconstructions, namely the conservative and the non-conservative ones, defined in the cells or on the edges. The same methodology was adapted for the bidimensional Stokes problem in [9] using a staggered discretization with polynomial reconstructions in the primal mesh for the pressure and in the associated diamond mesh for the velocity. We adapt such methodology to the Navier-Stokes and Euler systems for incompressible flows.

3.1 Stencils

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. A polynomial reconstruction of degree d requires $n_d = (d+1)(d+2)/2$ coefficients. In practice, the stencil is constituted of the N_d closest cells to each geometrical entity in the respective mesh, with $N_d \geq n_d$ (we usually consider $N_d \approx 1.5 n_d$ for the sake of robustness). In the present paper, we associate for each primal cell c_i, $i \in \mathscr{C}_{\mathscr{M}}$, diamond cell c_k, $k \in \mathscr{C}_{\mathscr{D}}$, and diamond edge $e_{k\ell}$, $k \in \mathscr{C}_{\mathscr{M}}$, $\ell \in \nu(k)$, the stencils S_i, consisting of indices of neighbor primal cells, and S_k and $S_{k\ell}$, consisting of indices of neighbor diamond cells, respectively.

3.2 Polynomial reconstruction

Let us consider a generic real-valued function ψ defined in Ω. To compute polynomial reconstructions based on the data of the associated stencil, we assume that we know the approximations of the mean value of ψ over the primal cells and the diamond cells, given by

$$\psi_i \approx \frac{1}{|c_i|}\int_{c_i} \psi \,\mathrm{d}x, \quad i \in \mathscr{C}_{\mathscr{M}}, \qquad \psi_k \approx \frac{1}{|c_k|}\int_{c_k} \psi \,\mathrm{d}x, \quad k \in \mathscr{C}_{\mathscr{D}}.$$

In what follows we use the usual conventions $\alpha = (\alpha_1, \alpha_2)$ with $|\alpha| = \alpha_1 + \alpha_2$ and $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2}$.

3.2.1 Conservative reconstruction for primal and diamond cells

For each primal cell c_i, $i \in \mathscr{C}_{\mathscr{M}}$, the local polynomial approximation of degree d of the underlying solution ψ is defined as

$$\boldsymbol{\psi}_i(x) = \psi_i + \sum_{1\le|\alpha|\le d} \mathscr{R}_i^\alpha \left[(x-m_i)^\alpha - M_i^\alpha\right],$$

where vector $\mathscr{R}_i = (\mathscr{R}_i^\alpha)_{1\le|\alpha|\le d}$ gathers the unknown 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

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

For a given stencil S_i, we consider the quadratic functional

$$E_i(\mathscr{R}_i) = \sum_{q\in S_i} \left[\frac{1}{|c_q|}\int_{c_q} \boldsymbol{\psi}_i(x) \,\mathrm{d}x - \psi_q\right]^2. \tag{10}$$

We denote by $\widehat{\mathscr{R}}_i$ the unique vector which minimizes quadratic functional (10) and we defined by $\widehat{\boldsymbol{\psi}}_i(x)$ the polynomial which corresponds to the best approximation in the least squares sense. A similar procedure is carried out when dealing with diamond cell c_k, $k \in \mathscr{C}_{\mathscr{D}}$.

3.2.2 Non-conservative reconstruction for inner diamond edges

For each inner diamond edge $e_{k\ell}$, $k \in \mathscr{C}_{\mathscr{D}}$, $\ell \in \nu(k)$, $\ell \neq \mathrm{D}$, the local polynomial approximation of degree d of the underlying function ψ is defined as

$$\boldsymbol{\psi}_{k\ell}(x) = \sum_{0 \leq |\alpha| \leq d} \mathscr{R}^{\alpha}_{k\ell}(x - m_{k\ell})^{\alpha},$$

where vector $\mathscr{R}_{k\ell} = (\mathscr{R}^{\alpha}_{k\ell})_{0 \leq |\alpha| \leq d}$ gathers the unknown 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 $\omega_{k\ell} = (\omega_{k\ell,q})_{q=1,\ldots,\#S_{k\ell}}$, $\omega_{k\ell,q}$ being the positive weights of the reconstruction, we consider the quadratic functional

$$E_{k\ell}(\mathscr{R}_{k\ell}) = \sum_{q \in S_{k\ell}} \omega_{k\ell,q} \left[\frac{1}{|c_q|} \int_{c_q} \boldsymbol{\psi}_{k\ell}(x)\, \mathrm{d}x - \psi_q \right]^2. \tag{11}$$

We denote by $\widetilde{\mathscr{R}}_{k\ell}$ the unique vector which minimizes the quadratic functional (11) and we define by $\widetilde{\boldsymbol{\psi}}_{k\ell}(x)$ the polynomial which corresponds to the best approximation in the least squares sense.

Remark 1 Introduction of weights is of crucial importance to provide stability. One has to set larger values for the adjacent cells of the edge as shown in [11].

3.2.3 Conservative reconstruction for boundary diamond edges

We treat the boundary diamond edges in a particular way in order to take into account the prescribed Dirichlet boundary condition. For each boundary diamond edge $e_{k\mathrm{D}}$, $k \in \mathscr{C}_{\mathscr{D}}$, the

local polynomial approximation of degree d of the underlying function ψ is defined as

$$\boldsymbol{\psi}_{k\mathrm{D}}(x) = \psi_{k\mathrm{D}} + \sum_{1\leq|\alpha|\leq d} \mathscr{R}^{\alpha}_{k\mathrm{D}} \left[(x - m_{k\mathrm{D}})^{\alpha} - M^{\alpha}_{k\mathrm{D}}\right],$$

where vector $\mathscr{R}_{k\mathrm{D}} = (\mathscr{R}^{\alpha}_{k\mathrm{D}})_{1\leq|\alpha|\leq d}$ gathers the unknown polynomial coefficients, $\psi_{k\mathrm{D}}$ is an approximation of order $2R$ of the mean value of function ψ_{D}, a generic real-valued function defined on $\partial\Omega$, 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

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

For a given stencil $S_{k\mathrm{D}}$, we consider the quadratic functional

$$E_{k\mathrm{D}}(\mathscr{R}_{k\mathrm{D}}) = \sum_{q\in S_{k\mathrm{D}}} \omega_{k\mathrm{D},q}\left[\frac{1}{|c_q|}\int_{c_q} \boldsymbol{\psi}_{k\mathrm{D}}(x)\,\mathrm{d}x - \psi_q\right]^2, \tag{12}$$

where $\omega_{k\mathrm{D}} = (\omega_{k\mathrm{D},q})_{q=1,\ldots,\#S_{i\mathrm{D}}}$ gather the positive weights of the reconstruction $\omega_{i\mathrm{D},q}$.

We denote by $\widehat{\mathscr{R}}_{k\mathrm{D}}$ the unique vector which minimizes the quadratic functional (12) and we define by $\widehat{\boldsymbol{\psi}}_{k\mathrm{D}}(x)$ the polynomial which corresponds to the best approximation in the least squares sense. The previous remark is also valid for this reconstruction.

3.3 Application of the polynomial reconstruction to the Navier-Stokes problem

We now introduce the polynomial reconstructions we shall use for U, Λ, and P to compute high-order approximations of the physical fluxes (notice that these variables require several kinds of reconstruction depending on their action in the equation). To this end, we assume that we know approximations to the mean values of functions U_1, U_2, Λ_1, and Λ_2 at the

diamond cells

$$U_{\beta,k} \approx \frac{1}{|c_k|}\int_{c_k} U_\beta \,\mathrm{d}x, \qquad \Lambda_{\beta,k} \approx \frac{1}{|c_k|}\int_{c_k} \Lambda_\beta \,\mathrm{d}x, \qquad \beta = 1,2,$$

and of function P at the primal cells

$$P_i \approx \frac{1}{|c_i|}\int_{c_i} P \,\mathrm{d}x,$$

that we gather in vectors $\boldsymbol{\Phi}^U = (U_{1,k}, U_{2,k})_{k=I+1,\dots,I+K} \in \mathbb{R}^{2K}$, $\boldsymbol{\Phi}^\Lambda = (\Lambda_{1,k}, \Lambda_{2,k})_{k=I+1,\dots,I+K} \in \mathbb{R}^{2K}$, and $\boldsymbol{\Phi}^P = (P_i)_{i=1,\dots,I} \in \mathbb{R}^I$. We this data at hand, we prescribe:

- for diamond cell c_k, $k \in \mathscr{C}_\mathscr{D}$, the conservative polynomial reconstructions $\widehat{\boldsymbol{U}}_k = (\widehat{\boldsymbol{U}}_{1,k}, \widehat{\boldsymbol{U}}_{2,k})$ taking the data in $\boldsymbol{\Phi}^U$ and $\widehat{\boldsymbol{\Lambda}}_k = (\widehat{\boldsymbol{\Lambda}}_{1,k}, \widehat{\boldsymbol{\Lambda}}_{2,k})$ taking the data in $\boldsymbol{\Phi}^\Lambda$ as given in section 3.2.1;
- for inner diamond edge $e_{k\ell}$, $k \in \mathscr{C}_\mathscr{D}$, $\ell \in \nu(k)$, $\ell \neq \mathrm{D}$, the non-conservative polynomial reconstructions $\widetilde{\boldsymbol{U}}_{k\ell} = (\widetilde{\boldsymbol{U}}_{1,k\ell}, \widetilde{\boldsymbol{U}}_{2,k\ell})$ taking the data in $\boldsymbol{\Phi}^U$ and $\widetilde{\boldsymbol{\Lambda}}_{k\ell} = (\widetilde{\boldsymbol{\Lambda}}_{1,k\ell}, \widetilde{\boldsymbol{\Lambda}}_{2,k\ell})$ taking the data in $\boldsymbol{\Phi}^\Lambda$ as given in section 3.2.2;
- for boundary diamond edge $e_{k\mathrm{D}}$, the conservative polynomial reconstruction $\widehat{\boldsymbol{U}}_{k\mathrm{D}} = (\widehat{\boldsymbol{U}}_{1,k\mathrm{D}}, \widehat{\boldsymbol{U}}_{2,k\mathrm{D}})$ taking the data in $\boldsymbol{\Phi}^U$ and the boundary condition, and $\widetilde{\boldsymbol{\Lambda}}_{k\mathrm{D}} = (\widetilde{\boldsymbol{\Lambda}}_{1,k\mathrm{D}}, \widetilde{\boldsymbol{\Lambda}}_{2,k\mathrm{D}})$ taking the data in $\boldsymbol{\Phi}^\Lambda$ as given in section 3.2.3;
- for primal cell c_i, $i \in \mathscr{C}_\mathscr{M}$, the conservative polynomial reconstruction $\widehat{\boldsymbol{P}}_i$ taking the data in $\boldsymbol{\Phi}^P$ as given in section 3.2.1.

In Table 1 we summarize all types of polynomial reconstructions we prescribe for each primal or diamond cell or edge. Notice that on the boundary edge $e_{k\mathrm{D}}$, we do not use the conservative reconstruction for Λ.

Table 1 Polynomial reconstructions for U, Λ, and P.

	c_k	$e_{k\ell}$	$e_{k\mathrm{D}}$	c_i
U	$\widehat{\boldsymbol{U}}_k = (\widehat{\boldsymbol{U}}_{1,k}, \widehat{\boldsymbol{U}}_{2,k})$	$\widetilde{\boldsymbol{U}}_{k\ell} = (\widetilde{\boldsymbol{U}}_{1,k\ell}, \widetilde{\boldsymbol{U}}_{2,k\ell})$	$\widehat{\boldsymbol{U}}_{k\mathrm{D}} = (\widehat{\boldsymbol{U}}_{1,k\mathrm{D}}, \widehat{\boldsymbol{U}}_{2,k\mathrm{D}})$	—
Λ	$\widehat{\boldsymbol{\Lambda}}_k = (\widehat{\boldsymbol{\Lambda}}_{1,k}, \widehat{\boldsymbol{\Lambda}}_{2,k})$	$\widetilde{\boldsymbol{\Lambda}}_{k\ell} = (\widetilde{\boldsymbol{\Lambda}}_{1,k\ell}, \widetilde{\boldsymbol{\Lambda}}_{2,k\ell})$	$\widetilde{\boldsymbol{\Lambda}}_{k\mathrm{D}} = (\widetilde{\boldsymbol{\Lambda}}_{1,k\mathrm{D}}, \widetilde{\boldsymbol{\Lambda}}_{2,k\mathrm{D}})$	—
P	—	—	—	$\widehat{\boldsymbol{P}}_i$

3.4 Flux evaluation

High-order approximations to the physical fluxes (7) are designed with the help of the polynomial reconstructions presented in the precious section.

3.4.1 Computation of the numerical fluxes

Now, assume that approximations $\Lambda_{k\ell,r}$ are known at each quadrature point $q_{k\ell,r}$, $k \in \mathscr{C}_{\mathscr{D}}$, $\ell \in \nu(k)$, $r = 1, \ldots, R$. We define the following fluxes:

– for an inner diamond edge $e_{k\ell}$, the physical fluxes (7) at the quadrature point $q_{k\ell,r}$ are approximated by

$$\mathscr{F}^{\otimes}_{\beta,k\ell,r} = \begin{cases} \widehat{\boldsymbol{U}}_{\beta,k}(q_{k\ell,r})(\Lambda_{k\ell,r} \cdot n_{k\ell}) & \text{if } \Lambda_{k\ell,r} \cdot n_{k\ell} \geq 0, \\ \widehat{\boldsymbol{U}}_{\beta,\ell}(q_{k\ell,r})(\Lambda_{k\ell,r} \cdot n_{k\ell}) & \text{otherwise,} \end{cases}$$

$$\mathscr{F}^{U}_{\beta,k\ell,r} = -\nu \nabla \widetilde{\boldsymbol{U}}_{\beta,k\ell}(q_{k\ell,r}) \cdot n_{k\ell}, \qquad \mathscr{F}^{P}_{\beta,k\ell,r} = \frac{1}{\rho} \widehat{\boldsymbol{P}}_i(q_{k\ell,r}) n_{\beta,k\ell},$$

respectively, with $\beta = 1, 2$ and the correspondence $i = \pi_{\mathscr{M}}(k, \ell)$;

– for a boundary diamond edge $e_{k\mathrm{D}}$, the physical fluxes (7) at the quadrature point $q_{k\mathrm{D},r}$ are approximated by

$$\mathscr{F}^{\otimes}_{\beta,k\mathrm{D},r} = \begin{cases} \widehat{\boldsymbol{U}}_{\beta,k}(q_{k\mathrm{D},r})(\Lambda_{k\mathrm{D},r}\cdot n_{k\mathrm{D}}) & \text{if } \Lambda_{k\mathrm{D},r}\cdot n_{k\mathrm{D}} \geq 0, \\ U_{\beta,\mathrm{D}}(q_{k\mathrm{D},r})(\Lambda_{k\mathrm{D},r}\cdot n_{k\mathrm{D}}) & \text{otherwise,} \end{cases}$$

$$\mathscr{F}^{U}_{\beta,k\mathrm{D},r} = -\nu\nabla\widehat{\boldsymbol{U}}_{\beta,k\mathrm{D}}(q_{k\mathrm{D},r})\cdot n_{k\mathrm{D}}, \qquad \mathscr{F}^{P}_{\beta,k\mathrm{D},r} = \frac{1}{\rho}\widehat{\boldsymbol{P}}_i(q_{k\mathrm{D},r})n_{\beta,k\mathrm{D}},$$

respectively, with $\beta = 1,2$ and the correspondence $i = \pi_{\mathscr{M}}(k,\mathrm{D})$;

– for an inner primal edge e_{ij}, the physical flux (9) at the quadrature point $q_{ij,r}$ is approximated by $\widehat{\boldsymbol{U}}_k(q_{ij,r})\cdot n_{ij}$, with the correspondence $k = \pi_{\mathscr{D}}(i,j)$;

– for a boundary primal edge $e_{i\mathrm{D}}$ the physical flux (9) at the quadrature point $q_{i\mathrm{D},r}$ is approximated by $U_{\mathrm{D}}(q_{i\mathrm{D},r})\cdot n_{i\mathrm{D}}$.

3.4.2 Computation of $\Lambda_{k\ell,r}$

To compute the approximations $\Lambda_{k\ell,r}$ at each quadrature point of each diamond edge, two scenarios are considered, namely, using a centred scheme and using upwind scheme that we will compare numerically. For the former scenario we set $\Lambda_{k\ell,r} = \widetilde{\boldsymbol{\Lambda}}_{k\ell}(q_{k\ell,r})$ while for the latter we set for inner diamond and boundary diamond edges, respectively,

$$\Lambda_{k\ell,r} = \begin{cases} \widehat{\boldsymbol{\Lambda}}_k(q_{k\ell,r}) & \text{if } \widetilde{\boldsymbol{\Lambda}}_{k\ell}(q_{k\ell,r})\cdot n_{k\ell} \geq 0, \\ \widehat{\boldsymbol{\Lambda}}_\ell(q_{k\ell,r}) & \text{otherwise,} \end{cases} \quad \text{and} \quad \Lambda_{k\mathrm{D},r} = \begin{cases} \widehat{\boldsymbol{\Lambda}}_k(q_{k\mathrm{D},r}) & \text{if } \widetilde{\boldsymbol{\Lambda}}_{k\mathrm{D}}(q_{k\mathrm{D},r})\cdot n_{k\mathrm{D}} \geq 0, \\ \widetilde{\boldsymbol{\Lambda}}_{k\mathrm{D}}(q_{k\mathrm{D},r}) & \text{otherwise.} \end{cases}$$

Remark 2 We highlight that the Dirichlet boundary conditions are not used to compute $\Lambda_{k\ell,r}$. Indeed, we report that the introduction of boundary condition in the guess velocity gives rise

of oscillations and prevent the iterative algorithm from converging. We shall present some numerical tests in order to emphasize such a problem.

4 Residual scheme and fixed-point algorithm

In this section let us rewrite the finite volume scheme in the residual form and introduce the fixed-point algorithm associated to the Navier-Stokes problem.

4.1 Residual operators for the linearized problem

For any vectors $\boldsymbol{\Phi} = (\boldsymbol{\Phi}^U, \boldsymbol{\Phi}^P) \in \mathbb{R}^{2K+I}$ and $\boldsymbol{\Psi} = \boldsymbol{\Phi}^\Lambda \in \mathbb{R}^{2K}$ we define the residual operators for the diamond mesh $\mathscr{C}_D$, as

$$\mathscr{G}^{\mathrm{NS}}_{\beta,k}(\boldsymbol{\Phi};\boldsymbol{\Psi}) = \sum_{\ell\in\nu(k)} |e_{k\ell}| \left[\sum_{r=1}^{R} \zeta_r \left(\mathscr{F}^{\otimes}_{\beta,k\ell,r} + \mathscr{F}^{U}_{\beta,k\ell,r} + \mathscr{F}^{P}_{\beta,k\ell,r}\right)\right] - |c_k| g_{\beta,k}, \qquad \beta = 1,2,$$

$$\mathscr{G}^{\mathrm{NS}}(\boldsymbol{\Phi};\boldsymbol{\Psi}) = \left(\mathscr{G}^{\mathrm{NS}}_{1,k}(\boldsymbol{\Phi};\boldsymbol{\Psi}),\, \mathscr{G}^{\mathrm{NS}}_{2,k}(\boldsymbol{\Phi};\boldsymbol{\Psi})\right)_{k=I+1,\ldots,I+K}$$

and for the primal mesh $\mathscr{C}_{\mathscr{M}}$, as

$$\mathscr{G}^{\nabla}_i(\boldsymbol{\Phi}) = \sum_{j\in\nu(i)} |e_{ij}| \left[\sum_{r=1}^{R} \zeta_r \mathscr{F}^{\nabla}_{ij,r}\right], \quad \mathscr{G}^{\nabla}(\boldsymbol{\Phi}) = (\mathscr{G}^{\nabla}_i(\boldsymbol{\Phi}))_{i=1,\ldots,I},$$

which correspond to the finite volume scheme (6) and (8) cast in residual form, respectively.

Gathering all the components of the residuals in vector $\mathscr{H}(\boldsymbol{\Phi};\boldsymbol{\Psi}) = (\mathscr{G}^{\mathrm{NS}}(\boldsymbol{\Phi};\boldsymbol{\Psi}), \mathscr{G}^{\nabla}(\boldsymbol{\Phi}))$, for a given vector $\boldsymbol{\Psi} \in \mathbb{R}^{2K}$, we introduce the global affine operator

$$\boldsymbol{\Phi} \in \mathbb{R}^{2K+I} \to \mathscr{H}(\boldsymbol{\Phi};\boldsymbol{\Psi}) \in \mathbb{R}^{2K+I}.$$

Vector $\Phi^\star = (\Phi^{U,\star}, \Phi^{P,\star}) \in \mathbb{R}^{2K+I}$ is the solution of the problem $\mathscr{H}(\Phi;\Psi) = 0$, and provides constant piecewise approximations of U_1, U_2, and P. We then build the nonlinear operator $\Psi \in \mathbb{R}^{2K} \to \Phi^{U,\star} = \mathbb{H}(\Psi) \in \mathbb{R}^{2K}$ which maps the guessed velocity to the solution of the Oseen equation.

4.2 Fixed-point algorithm

Based on the affine operator $\mathbb{H}(\Psi)$ defined above, we introduce a fixed-point algorithm to find the solution of the nonlinear problem $\mathbb{H}(\Phi^U) = \Phi^U$. The algorithm is described hereby and a schematic representation is shown in Fig. 3:

(1) Give $C \in \mathbb{R} \setminus \{0\}$ and set $\Psi^0 = (C) \in \mathbb{R}^{2K}$ (in practice we choose $C = 1$ for the sake of convenience), give $\mathrm{R_{MAX}} \in \mathbb{R}^+$, and set $n = 0$;

(2) Compute $\Phi^{U,n} = \mathbb{H}(\Psi^n)$ and $\Phi^{P,n}$ solution of $\mathscr{H}(\Phi;\Psi^n) = 0$;

(3) Compute the L^1-norm residual R^n, given by $R^n = \|\Phi^{U,n} - \Psi^n\| / \|\Psi^n\|$, where

$$\|\Phi^{U,n} - \Psi^n\| = \sum_{\beta=1}^{2} \frac{\sum\limits_{k \in \mathscr{C}_\mathscr{D}} |U^n_{\beta,k} - \Lambda^n_{\beta,k}||c_k|}{\sum\limits_{k \in \mathscr{C}_\mathscr{D}} |c_k|}, \qquad \|\Psi^n\| = \sum_{\beta=1}^{2} \frac{\sum\limits_{k \in \mathscr{C}_\mathscr{D}} |\Lambda^n_{\beta,k}||c_k|}{\sum\limits_{k \in \mathscr{C}_\mathscr{D}} |c_k|};$$

(4) If $R^n < \mathrm{R_{MAX}}$, stop the algorithm and $\Phi^{U,\star} = \Phi^{U,n}$ is the solution; otherwise go to (5);

(5) Compute Ψ^{n+1} as a relaxation between $\Phi^{U,n}$ and Ψ^n, with θ the relaxation coefficient, as

$$\Psi^{n+1} = \theta \Phi^{U,n} + (1-\theta)\Psi^n;$$

(6) Increment n and go to (2).

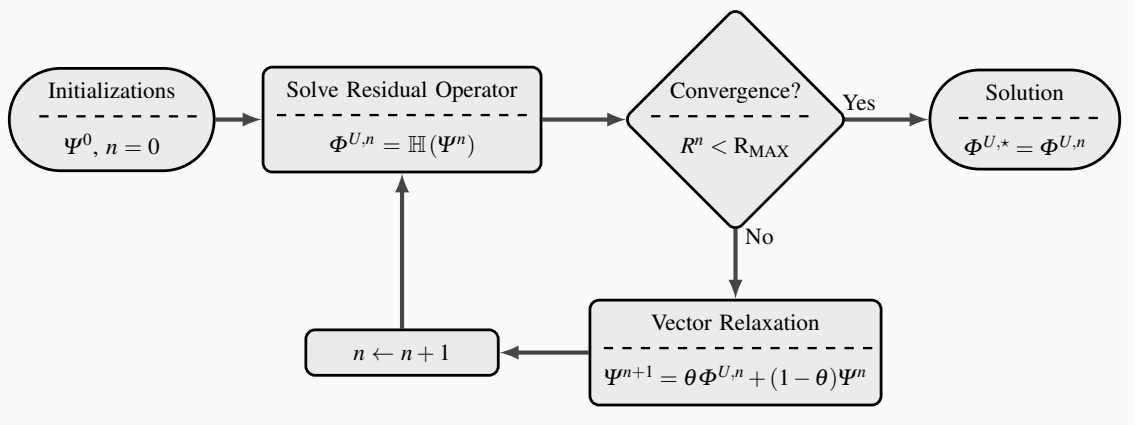

Fig. 3 Fixed-point algorithm to find the fixed-point of operator $\mathbb{H}$.

Remark 3 We highlight that if one sets $\Lambda^n = 0$ when dealing with the Euler problem ($\nu = 0$), the resulting system of partial differential equations is $\nabla \cdot (0 \otimes U + PI_2) = f$ and $\nabla \cdot U = 0$. Clearly the first equation becomes $\nabla \cdot (PI_2) = f$ and the problem admits and infinity number of solutions for U. In the same way, it is not wise to choose $\Psi^0 = 0$ since the fixed-point algorithm will never converge.

5 Synthetic benchmark

In this section we quantitatively and qualitatively assess the robustness and accuracy of the proposed method as well as its efficiency to solve the incompressible Navier-Stokes and Euler systems of PDEs. The numerical tests were performed considering three situations: an inviscid fluid (leading to the Euler problem), a fluid with viscosity $\nu = 10^{-5}$, and an ultra-viscous fluid with $\nu = 1$.

In order to validate the implementation of the method and assess the convergence rates, we manufacture an analytical solution on the unit square domain $\Omega =]0,1[^2$, setting

$$U_1(x) = \frac{1}{2}(3 - \cos(\pi x_1)\sin(\pi x_2)), \qquad U_2(x) = \frac{1}{2}(3 - \sin(\pi x_1)\cos(\pi x_2)),$$

$$P(x) = \frac{1}{4}(\cos(2\pi x_1) + \cos(2\pi x_2)).$$

We plot in Fig. 4 the isocontours of the horizontal component of the velocity, U_1, and of the vertical component of the velocity, U_2, the isocontours of the magnitude of the velocity, $||U||$, and the isocontours of the pressure, P.

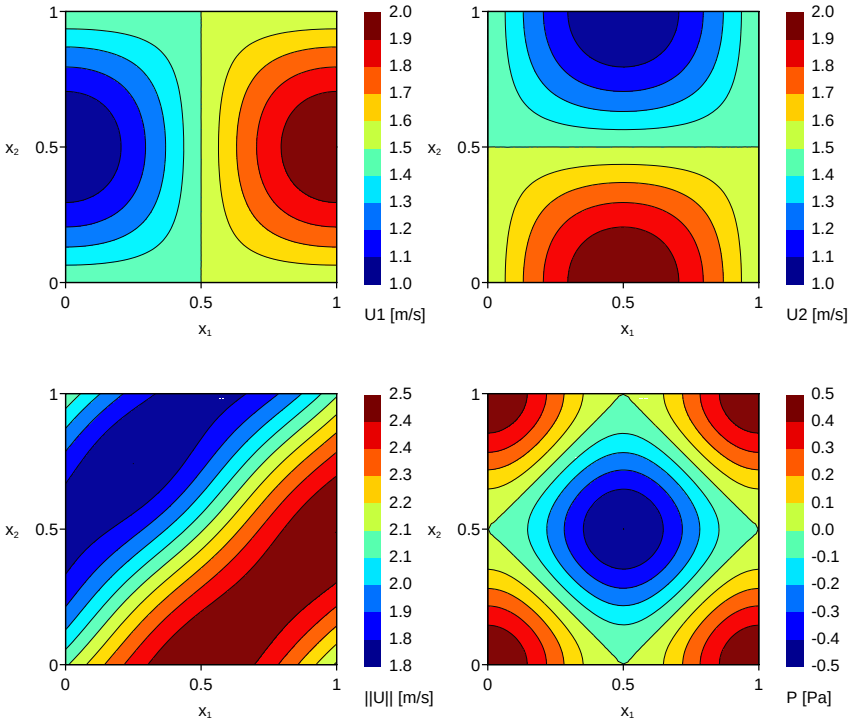

Fig. 4 Isocontours of the horizontal component (top, left) and the vertical component of the velocity (top, right), isocontours of the velocity magnitude (bottom, left), and isocontours of the pressure (bottom, right).

The source terms are computed such that equations (1-2) hold and the Dirichlet boundary condition derives from the exact solution.

5.1 Accuracy and error convergence rates assessment

To assess the accuracy and the convergence of the proposed method, consider vectors $\Phi_\beta^{U,\star} = (U_{\beta,k}^\star)_{k\in\mathscr{C}_\mathscr{D}}$, $\beta = 1,2$ and $\Phi^{P,\star} = (P_i^\star)_{i\in\mathscr{C}_\mathscr{M}}$ which gather the numerical approximations of the mean values of U_β, $\beta = 1,2$, and P, respectively, and vectors $\overline{\Phi}_\beta^U = (\overline{U}_{\beta,k})_{k\in\mathscr{C}_\mathscr{D}}$, $\beta = 1,2$, and $\overline{\Phi}^P = (\overline{P}_i)_{i\in\mathscr{C}_\mathscr{M}}$ which gather the exact mean values $\overline{U}_{\beta,k}$ and $\overline{P}_i$ of U_β and P, respectively.

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

The L^1-norm errors are given by

$$E_{\beta,1}^U(\mathscr{D}) = \sum_{k\in\mathscr{C}_\mathscr{D}} |U_{\beta,k}^\star - \overline{U}_{\beta,k}||c_k|, \qquad E_1^P(\mathscr{M}) = \sum_{i\in\mathscr{C}_\mathscr{M}} |P_i^\star - \overline{P}_i||c_i|,$$

and the L^∞-norm errors are given by

$$E_{\beta,\infty}^U(\mathscr{D}) = \max_{k\in\mathscr{C}_\mathscr{D}} |U_{\beta,k}^\star - \overline{U}_{\beta,k}|, \qquad E_\infty^P(\mathscr{M}) = \max_{i\in\mathscr{C}_\mathscr{M}} |P_i^\star - \overline{P}_i|.$$

The convergence rate for P with the L^1-norm errors between two different primal meshes $\mathscr{M}_1$ and $\mathscr{M}_2$, with I_1 and I_2 cells, respectively, where $I_1 \neq I_2$, is evaluated as

$$O_1(\mathscr{M}_1,\mathscr{M}_2) = 2\frac{|\log(E_1^P(\mathscr{M}_1)/E_1^P(\mathscr{M}_2))|}{|\log(I_1/I_2)|}.$$

Likewise for diamond meshes $\mathscr{D}_1$ and $\mathscr{D}_2$ with K_1 and K_2 cells, we have

$$O_1(\mathscr{D}_1,\mathscr{D}_2) = 2\frac{|\log(E^U_{\beta,1}(\mathscr{D}_1)/E^U_{\beta,1}(\mathscr{D}_2))|}{|\log(K_1/K_2)|}, \quad \beta = 1,2.$$

We perform a similar estimate for the norm L^∞ by substituting O_1 with O_∞.

In all the simulations the weights in functionals (11-12) are set to $\omega_{ij,q} = 5$, $i \in \mathscr{C}_\mathscr{M}$, $j \in \nu(i)$, $q \in S_{ij}$, if e_{ij} is an edge of c_q and $\omega_{ij,q} = 1$, otherwise. In the same way, for the reconstructions on diamond edges we set $\omega_{\beta,k\ell,q} = 5$, $k \in \mathscr{C}_\mathscr{D}$, $\ell \in \nu(k)$, $q \in S_{k\ell}$, $\beta = 1,2$, if $e_{k\ell}$ is an edge of c_q and $\omega_{\beta,k\ell,q} = 1$, otherwise [11].

The choice of θ does not change the errors and convergence orders and $\theta = 0.5$ was employed for this benchmark.

5.1.1 Inviscid fluid

Simulations were carried out with successively finer regular triangular Delaunay meshes (see Fig. 5, left) and the associated diamond meshes (see Fig. 5, right). The goal is to measure the order of convergence of the method.

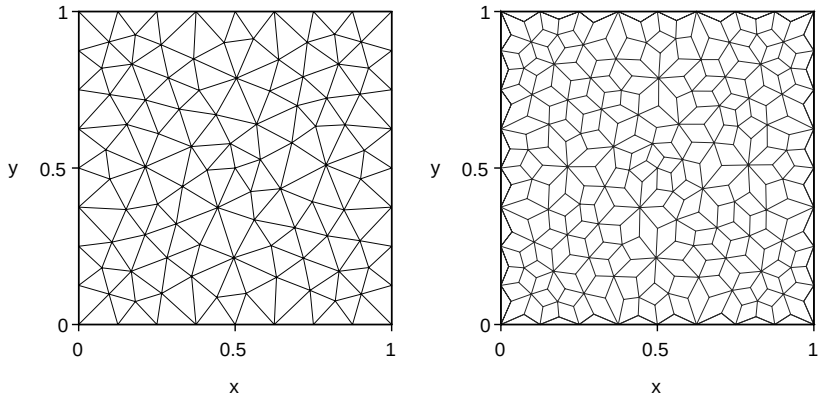

Fig. 5 A coarse uniform triangular Delaunay mesh (left) and the associated diamond mesh (right).

In section 3.4.2, two reconstructions for the linearized velocity $\Lambda_{k\ell,r}$ are proposed. For the centred approximations, Tables 2 and 3 give the L^1- and L^∞-norm errors and the convergence rates using $\mathbb{P}_1$, $\mathbb{P}_3$, 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 $\mathbb{P}_1$ polynomial reconstruction provides a second-order approximation both for U_β, $\beta = 1,2$, and for P while the $\mathbb{P}_3$ reconstruction achieves an effective fourth-order approximation and the $\mathbb{P}_5$ reconstruction a sixth-order approximation. The convergence rates are valid both for the L^1- and L^∞-norm errors and we highlight that no oscillations or numerical locking are reported in all the experiences.

Table 2 L^1-norm error and convergence rates for $\nu = 0$ with centred $\Lambda_{k\ell,r}$ approximations and using uniform triangular Delaunay primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_1	O_1	E_1	O_1	E_1	O_1
U_1	168	1.06E−02	—	2.22E−03	—	4.62E−04	—
	585	3.03E−03	2.01	1.40E−04	4.43	1.28E−05	5.75
	2373	7.28E−04	2.04	8.82E−06	3.95	1.90E−07	6.01
	9339	1.91E−04	1.95	5.61E−07	4.02	4.10E−09	5.60
U_2	168	9.87E−03	—	2.29E−03	—	4.75E−04	—
	585	2.85E−03	1.99	1.45E−04	4.42	1.31E−05	5.76
	2373	7.26E−04	1.95	9.04E−06	3.96	1.94E−07	6.01
	9339	1.89E−04	1.96	5.60E−07	4.06	4.11E−09	5.63
P	104	2.03E−02	—	4.24E−03	—	7.30E−04	—
	374	4.98E−03	2.20	2.63E−04	4.34	1.68E−05	5.89
	1550	1.10E−03	2.13	1.49E−05	4.04	1.72E−07	6.45
	6162	2.60E−04	2.09	9.15E−07	4.04	3.26E−09	5.75

Table 3 L^∞-norm error and convergence rates for $\nu = 0$ with centred $\Lambda_{k\ell,r}$ approximations and using uniform triangular Delaunay primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_∞	O_∞	E_∞	O_∞	E_∞	O_∞
U_1	168	5.07E−02	—	6.98E−03	—	1.79E−03	—
	585	2.43E−02	1.18	5.98E−04	3.94	5.84E−05	5.49
	2373	5.13E−03	2.22	5.01E−05	3.54	1.29E−06	5.45
	9339	2.17E−03	1.26	4.53E−06	3.51	1.88E−08	6.17
U_2	168	6.92E−02	—	8.77E−03	—	1.75E−03	—
	585	1.20E−02	2.81	7.76E−04	3.89	5.03E−05	5.69
	2373	4.33E−03	1.46	5.33E−05	3.83	1.06E−06	5.51
	9339	2.22E−03	0.98	4.69E−06	3.55	1.66E−08	6.07
P	104	8.86E−02	—	1.33E−02	—	2.37E−03	—
	374	2.37E−02	2.06	1.01E−03	4.02	1.35E−04	4.48
	1550	7.71E−03	1.58	6.66E−05	3.83	2.13E−06	5.84
	6162	2.15E−03	1.85	5.11E−06	3.72	3.24E−08	6.07

Robustness and accuracy assessments under deformed meshes are of crucial importance in order to check the method capability to handle complex meshes still preserving high-order convergence rates. To this end, we consider successively finer deformed quadrilateral meshes (see Fig. 6), obtained from a random displacement of each inner vertex of structured meshes controlled by a deformation factor (see the detailed procedure in [11]). In the present experience, we choose a test case with 30% of deformation (see Fig. 6).

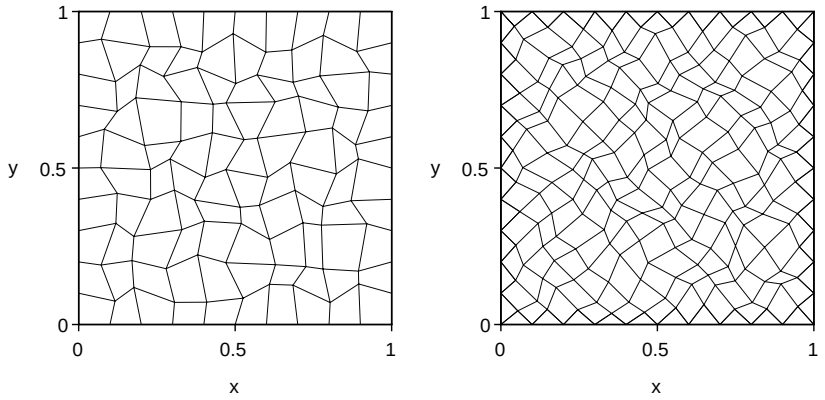

Fig. 6 A coarse deformed quadrilateral mesh (left) and the associated diamond mesh (right).

We deal again with the centred approximations for $\Lambda_{k\ell,r}$ and report in Tables 4 and 5 the L^1- and L^∞-norm errors and convergence rates using $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, respectively. The scheme correctly handles complex meshes and the convergence rates are optimal both for velocity and pressure and for both norm errors. Up to sixth-order convergence rates are achieved using the $\mathbb{P}_5$ polynomial reconstruction and no oscillations or numerical locking were noticed in all the experiences.

Table 4 L^1-norm error and convergence rates for $\nu = 0$ with centred $\Lambda_{k\ell,r}$ approximations and using deformed quadrilateral primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_1	O_1	E_1	O_1	E_1	O_1
U_1	168	1.06E−02	—	2.38E−03	—	1.03E−03	—
	585	3.03E−03	2.01	1.77E−04	4.17	1.36E−05	6.93
	2373	7.28E−04	2.04	1.71E−05	3.33	2.07E−07	5.98
	9339	1.91E−04	1.95	1.07E−06	4.04	3.95E−09	5.78
U_2	168	9.87E−03	—	2.24E−03	—	9.65E−04	—
	585	2.85E−03	1.99	1.86E−04	3.99	1.34E−05	6.85
	2373	7.26E−04	1.95	1.68E−05	3.43	2.10E−07	5.93
	9339	1.89E−04	1.96	1.06E−06	4.03	3.95E−09	5.80
P	104	2.03E−02	—	3.83E−03	—	1.65E−03	—
	374	4.98E−03	2.20	3.52E−04	3.73	1.64E−05	7.21
	1550	1.10E−03	2.13	3.18E−05	3.38	2.16E−07	6.09
	6162	2.60E−04	2.09	1.75E−06	4.20	3.29E−09	6.06

Table 5 L^∞-norm error and convergence rates for $\nu = 0$ with centred $\Lambda_{k\ell,r}$ approximations and using deformed quadrilateral primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_∞	O_∞	E_∞	O_∞	E_∞	O_∞
U_1	168	5.07E−02	—	5.09E−03	—	2.48E−03	—
	585	2.43E−02	1.18	7.35E−04	3.10	6.67E−05	5.80
	2373	5.13E−03	2.22	4.97E−05	3.85	1.12E−06	5.83
	9339	2.17E−03	1.26	4.52E−06	3.50	2.34E−08	5.65
U_2	168	6.92E−02	—	7.56E−03	—	2.59E−03	—
	585	1.20E−02	2.81	9.41E−04	3.34	7.08E−05	5.77
	2373	4.33E−03	1.46	6.85E−05	3.74	1.03E−06	6.05
	9339	2.22E−03	0.98	6.38E−06	3.47	1.74E−08	5.96
P	104	8.86E−02	—	1.50E−02	—	5.42E−03	—
	374	2.37E−02	2.06	1.34E−03	3.77	1.48E−04	5.63
	1550	7.71E−03	1.58	1.46E−04	3.12	2.34E−06	5.84
	6162	2.15E−03	1.85	9.41E−06	3.97	5.37E−08	5.47

We repeat the numerical tests considering now the upwind approximation for $\Lambda_{k\ell,r}$. For regular triangular Delaunay meshes, we report in Tables 6 and 7 the L^1- and L^∞-norm errors, respectively, and the associated convergence rates using the $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions. The scheme based on the upwind approximations of $\Lambda_{k\ell,r}$ provides the expected convergence orders for both the velocity and the pressure, either for the L^1- or L^∞-norm errors. The values are very similar to the centred case.

Table 6 L^1-norm error and convergence rates for $\nu = 0$ with upwind $\Lambda_{k\ell,r}$ approximations and using uniform triangular Delaunay primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_1	O_1	E_1	O_1	E_1	O_1
U_1	168	1.63E−02	—	2.70E−03	—	5.16E−04	—
	585	3.87E−03	2.30	1.16E−04	5.05	1.33E−05	5.86
	2373	9.72E−04	1.97	8.76E−06	3.69	2.10E−07	5.92
	9339	2.87E−04	1.78	5.33E−07	4.09	3.48E−09	5.98
U_2	168	1.55E−02	—	2.70E−03	—	5.15E−04	—
	585	3.86E−03	2.23	1.20E−04	5.00	1.35E−05	5.84
	2373	9.64E−04	1.98	8.89E−06	3.72	2.13E−07	5.92
	9339	2.87E−04	1.77	5.32E−07	4.11	3.50E−09	6.00
P	104	3.33E−02	—	4.59E−03	—	8.77E−04	—
	374	8.13E−03	2.20	2.20E−04	4.75	1.75E−05	6.11
	1550	1.96E−03	2.00	1.45E−05	3.83	1.62E−07	6.59
	6162	5.04E−04	1.97	8.83E−07	4.06	2.38E−09	6.11

Table 7 L^∞-norm error and convergence rates for $\nu = 0$ with upwind $\Lambda_{k\ell,r}$ approximations and using uniform triangular Delaunay primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_∞	O_∞	E_∞	O_∞	E_∞	O_∞
U_1	168	9.22E−02	—	8.56E−03	—	1.96E−03	—
	585	2.59E−02	2.04	4.97E−04	4.56	5.45E−05	5.74
	2373	5.72E−03	2.16	4.60E−05	3.40	1.19E−06	5.46
	9339	3.17E−03	0.86	4.18E−06	3.50	1.92E−08	6.02
U_2	168	4.76E−02	—	7.58E−03	—	2.09E−03	—
	585	3.35E−02	0.56	5.86E−04	4.10	4.12E−05	6.30
	2373	6.18E−03	2.41	4.23E−05	3.75	1.07E−06	5.22
	9339	3.39E−03	0.88	4.26E−06	3.35	1.88E−08	5.90
P	104	1.63E−01	—	1.55E−02	—	2.90E−03	—
	374	3.80E−02	2.28	9.03E−04	4.44	1.24E−04	4.93
	1550	1.26E−02	1.55	7.42E−05	3.52	2.02E−06	5.79
	6162	3.74E−03	1.76	5.25E−06	3.84	3.99E−08	5.69

In Tables 8 and 9 are gathered the L^1 and L^∞ results in the case of successively finer deformed quadrilateral meshes. Once again, the accuracy of the numerical approximations is comparable to the centred case. The scheme is robust and stable when dealing with complex meshes and provides the expected convergence rates for all variables.

Table 8 L^1-norm error and convergence rates for $\nu = 0$ with upwind $\Lambda_{k\ell,r}$ approximations and using deformed quadrilateral primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_1	O_1	E_1	O_1	E_1	O_1
U_1	168	1.63E−02	—	2.30E−03	—	9.87E−04	—
	585	3.87E−03	2.30	1.74E−04	4.14	1.17E−05	7.11
	2373	9.72E−04	1.97	1.81E−05	3.24	1.64E−07	6.09
	9339	2.87E−04	1.78	1.08E−06	4.12	3.23E−09	5.73
U_2	168	1.55E−02	—	2.28E−03	—	9.39E−04	—
	585	3.86E−03	2.23	1.83E−04	4.05	1.15E−05	7.06
	2373	9.64E−04	1.98	1.76E−05	3.34	1.66E−07	6.05
	9339	2.87E−04	1.77	1.07E−06	4.09	3.25E−09	5.74
P	104	3.33E−02	—	4.00E−03	—	1.66E−03	—
	374	8.13E−03	2.20	3.36E−04	3.87	1.38E−05	7.48
	1550	1.96E−03	2.00	3.37E−05	3.23	1.62E−07	6.25
	6162	5.04E−04	1.97	1.78E−06	4.26	2.63E−09	5.97

Table 9 L^∞-norm error and convergence rates for $\nu = 0$ with upwind $\Lambda_{k\ell,r}$ approximations and using deformed quadrilateral primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_∞	O_∞	E_∞	O_∞	E_∞	O_∞
U_1	168	9.22E−02	—	5.42E−03	—	2.17E−03	—
	585	2.59E−02	2.04	6.14E−04	3.49	6.20E−05	5.70
	2373	5.72E−03	2.16	5.04E−05	3.57	1.08E−06	5.78
	9339	3.17E−03	0.86	4.69E−06	3.47	2.13E−08	5.73
U_2	168	4.76E−02	—	6.23E−03	—	2.25E−03	—
	585	3.35E−02	0.56	7.32E−04	3.43	6.32E−05	5.73
	2373	6.18E−03	2.41	6.42E−05	3.48	1.12E−06	5.76
	9339	3.39E−03	0.88	4.99E−06	3.73	2.07E−08	5.83
P	104	1.63E−01	—	1.41E−02	—	5.43E−03	—
	374	3.80E−02	2.28	1.18E−03	3.87	1.37E−04	5.75
	1550	1.26E−02	1.55	1.56E−04	2.85	1.77E−06	6.12
	6162	3.74E−03	1.76	9.73E−06	4.02	5.42E−08	5.05

5.1.2 Viscous fluid with $\nu = 10^{-5}$

We turn now to the case of a viscous fluid with $\nu = 10^{-5}$ and present the results obtained with the centred version for $\Lambda_{k\ell,r}$ since both approximations (centred and upwind) achieved comparable accuracy when no viscous term is present.

Tables 10 and 11 provide L^1- and L^∞-norm errors and the convergence rates using $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, respectively, using regular triangular Delaunay meshes as in Fig. 5. We obtain effective second-, fourth-, and sixth-order convergence rates for the velocity and the pressure.l reconstructions, respectively. The convergence rates are the expected ones both for L^1- and L^∞-norm errors and no oscillations or numerical locking are reported in all the experiences.

Table 10 L^1-norm error and convergence rates for $\nu = 10^{-5}$ with centred $\Lambda_{k\ell,r}$ approximations and using uniform triangular Delaunay primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_1	O_1	E_1	O_1	E_1	O_1
U_1	168	1.06E−02	—	2.22E−03	—	4.61E−04	—
	585	3.03E−03	2.01	1.40E−04	4.44	1.28E−05	5.75
	2373	7.27E−04	2.04	8.81E−06	3.95	1.89E−07	6.02
	9339	1.90E−04	1.96	5.58E−07	4.03	4.04E−09	5.61
U_2	168	9.86E−03	—	2.29E−03	—	4.74E−04	—
	585	2.85E−03	1.99	1.45E−04	4.42	1.31E−05	5.76
	2373	7.25E−04	1.95	9.02E−06	3.97	1.93E−07	6.02
	9339	1.88E−04	1.97	5.57E−07	4.07	4.05E−09	5.64
P	104	2.03E−02	—	4.24E−03	—	7.30E−04	—
	374	4.97E−03	2.20	2.63E−04	4.35	1.68E−05	5.89
	1550	1.09E−03	2.13	1.49E−05	4.04	1.71E−07	6.45
	6162	2.59E−04	2.08	9.11E−07	4.05	3.22E−09	5.76

Table 11 L^∞-norm error and convergence rates for $\nu = 10^{-5}$ with centred $\Lambda_{k\ell,r}$ approximations and using uniform triangular Delaunay primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_∞	O_∞	E_∞	O_∞	E_∞	O_∞
U_1	168	5.07E−02	—	6.98E−03	—	1.79E−03	—
	585	2.43E−02	1.18	5.97E−04	3.94	5.82E−05	5.49
	2373	5.12E−03	2.22	5.01E−05	3.54	1.28E−06	5.46
	9339	2.15E−03	1.27	4.53E−06	3.51	1.86E−08	6.17
U_2	168	6.92E−02	—	8.76E−03	—	1.74E−03	—
	585	1.20E−02	2.81	7.75E−04	3.89	5.03E−05	5.68
	2373	4.31E−03	1.46	5.31E−05	3.83	1.05E−06	5.52
	9339	2.21E−03	0.98	4.69E−06	3.54	1.63E−08	6.08
P	104	8.86E−02	—	1.33E−02	—	2.37E−03	—
	374	2.37E−02	2.06	1.01E−03	4.03	1.34E−04	4.49
	1550	7.71E−03	1.58	6.65E−05	3.83	2.12E−06	5.83
	6162	2.14E−03	1.85	5.10E−06	3.72	3.20E−08	6.08

We carry out the same simulations using deformed quadrilateral meshes as in Fig. 6. In Tables 12 and 13 we report the L^1- and L^∞- norm errors and convergence rates for $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, respectively. As in the invicid case, the method handles complex meshes since we obtain the expected convergence orders for both L^1- and L^∞-norm errors.

Table 12 L^1-norm error and convergence rates for $\nu = 10^{-5}$ with centred $\Lambda_{k\ell,r}$ approximations and using deformed quadrilateral primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_1	O_1	E_1	O_1	E_1	O_1
U_1	220	6.46E−03	—	9.93E−04	—	1.04E−03	—
	840	1.58E−03	2.10	9.79E−05	3.46	1.47E−05	6.36
	3280	3.82E−04	2.09	5.30E−06	4.28	3.79E−07	5.37
	12960	9.40E−05	2.04	3.18E−07	4.10	5.19E−09	6.25
U_2	220	5.42E−03	—	1.08E−03	—	1.08E−03	—
	840	1.48E−03	1.94	9.84E−05	3.58	1.52E−05	6.37
	3280	3.61E−04	2.07	5.52E−06	4.23	4.15E−07	5.29
	12960	9.03E−05	2.02	3.23E−07	4.13	5.45E−09	6.31
P	100	1.46E−02	—	3.40E−03	—	3.28E−03	—
	400	3.09E−03	2.24	1.61E−04	4.40	1.62E−05	7.66
	1600	6.87E−04	2.17	6.91E−06	4.54	4.03E−07	5.33
	6400	1.55E−04	2.15	2.86E−07	4.59	3.11E−09	7.02

Table 13 L^∞-norm error and convergence rates for $\nu = 10^{-5}$ with centred $\Lambda_{k\ell,r}$ approximations and using deformed quadrilateral primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_∞	O_∞	E_∞	O_∞	E_∞	O_∞
U_1	220	3.05E−02	—	4.28E−03	—	5.79E−03	—
	840	1.17E−02	1.43	8.34E−04	2.44	1.21E−04	5.77
	3280	3.71E−03	1.69	5.82E−05	3.91	1.48E−05	3.08
	12960	1.04E−03	1.85	4.49E−06	3.73	8.11E−08	7.58
U_2	220	2.26E−02	—	1.01E−02	—	1.46E−02	—
	840	8.79E−03	1.41	7.92E−04	3.80	1.80E−04	6.56
	3280	2.53E−03	1.83	7.61E−05	3.44	1.76E−05	3.41
	12960	7.42E−04	1.79	5.50E−06	3.83	8.27E−08	7.80
P	100	1.25E−01	—	5.20E−03	—	6.94E−03	—
	400	3.35E−02	1.90	5.73E−04	3.18	2.37E−04	4.87
	1600	9.32E−03	1.85	4.20E−05	3.77	1.05E−05	4.49
	6400	2.00E−03	2.22	4.56E−06	3.20	1.55E−07	6.08

5.1.3 Viscous fluid with $\nu = 1$

The last accuracy test concerns an ultra-viscous fluid with $\nu = 1$. As in the previous case, the following results were carried out only with centred approximations for $\Lambda_{k\ell,r}$. We have however checked that the upwind case provides comparable results. We report in Tables 14 and 15 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, and regular triangular Delaunay meshes. We observe an effective second-, fourth- and sixth-order convergence rates for the velocity with $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, respectively. Unlike to the other two cases (inviscid fluid and viscous fluid with $\nu = 10^{-5}$), we just achieved a first-, third-, and fifth-order convergence rates for the pressure with $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, respectively. Such results are expected when the diffusive term of the Navier-Stokes equation is dominant over the convective term or when no convective term is considered at all as in the the Stokes problem. The convergence rates are valid both for the L^1- and L^∞-norm errors and we highlight that no oscillations or numerical locking are reported in all the experiences.

Table 14 L^1-norm error and convergence rates for $\nu = 1$ with centred $\Lambda_{k\ell,r}$ approximations and using uniform triangular Delaunay primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_1	O_1	E_1	O_1	E_1	O_1
U_1	168	2.32E−03	—	1.88E−04	—	2.78E−05	—
	585	4.85E−04	2.51	9.54E−06	4.78	1.41E−07	8.47
	2373	1.14E−04	2.06	5.05E−07	4.20	1.50E−09	6.49
	9339	2.75E−05	2.08	3.02E−08	4.11	2.36E−11	6.06
U_2	168	2.51E−03	—	1.81E−04	—	2.65E−05	—
	585	5.08E−04	2.56	9.72E−06	4.69	1.31E−07	8.51
	2373	1.14E−04	2.13	5.09E−07	4.21	1.50E−09	6.38
	9339	2.74E−05	2.08	3.02E−08	4.12	2.29E−11	6.11
P	104	6.77E−02	—	5.27E−03	—	6.38E−04	—
	374	2.86E−02	1.35	4.10E−04	3.99	5.82E−06	7.34
	1550	1.14E−02	1.29	3.88E−05	3.32	9.87E−08	5.74
	6162	5.36E−03	1.09	4.25E−06	3.21	2.46E−09	5.35

Table 15 L^∞-norm error and convergence rates for $\nu = 1$ with centred $\Lambda_{k\ell,r}$ approximations and using uniform triangular Delaunay primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_∞	O_∞	E_∞	O_∞	E_∞	O_∞
U_1	168	9.41E−03	—	8.55E−04	—	2.39E−04	—
	585	3.45E−03	1.61	5.32E−05	4.45	1.98E−06	7.69
	2373	8.58E−04	1.99	3.79E−06	3.77	1.17E−08	7.33
	9339	2.17E−04	2.01	2.28E−07	4.10	1.41E−10	6.45
U_2	168	9.35E−03	—	6.99E−04	—	2.12E−04	—
	585	3.68E−03	1.49	4.30E−05	4.47	1.72E−06	7.72
	2373	8.79E−04	2.05	3.75E−06	3.48	1.02E−08	7.32
	9339	2.20E−04	2.02	2.23E−07	4.12	1.21E−10	6.47
P	104	2.34E−01	—	1.82E−02	—	5.01E−03	—
	374	1.37E−01	0.84	2.73E−03	2.96	7.29E−05	6.61
	1550	7.79E−02	0.80	3.88E−04	2.75	6.67E−07	6.60
	6162	3.59E−02	1.12	4.63E−05	3.08	2.76E−08	4.61

For the deformed quadrilateral meshes, we report in Tables 16 and 17 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 results confirm that the method can handle complex meshes still obtaining the expected convergence orders (the ones achieved using Delaunay meshes) for both L^1- and L^∞-norm errors.

Table 16 L^1-norm error and convergence rates for $\nu = 1$ with centred $\Lambda_{k\ell,r}$ approximations and using deformed quadrilateral primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_1	O_1	E_1	O_1	E_1	O_1
U_1	220	1.62E−03	—	5.35E−05	—	6.64E−06	—
	840	3.95E−04	2.11	2.43E−06	4.62	8.65E−08	6.48
	3280	8.99E−05	2.17	1.36E−07	4.24	5.28E−10	7.49
	12960	2.21E−05	2.04	8.25E−09	4.08	5.57E−12	6.63
U_2	220	1.96E−03	—	5.51E−05	—	6.90E−06	—
	840	3.97E−04	2.38	2.31E−06	4.74	9.33E−08	6.42
	3280	9.10E−05	2.16	1.42E−07	4.09	4.88E−10	7.71
	12960	2.19E−05	2.07	8.47E−09	4.10	5.76E−12	6.46
P	100	4.25E−02	—	1.37E−03	—	9.84E−05	—
	400	1.44E−02	1.56	1.33E−04	3.37	1.86E−06	5.72
	1600	5.28E−03	1.45	1.09E−05	3.61	2.72E−08	6.09
	6400	2.19E−03	1.27	1.06E−06	3.37	5.33E−10	5.67

Table 17 L^∞-norm error and convergence rates for $\nu = 1$ with centred $\Lambda_{k\ell,r}$ approximations and using deformed quadrilateral primal meshes.

		$\mathbb{P}_1$		$\mathbb{P}_3$		$\mathbb{P}_5$	
	DOF	E_∞	O_∞	E_∞	O_∞	E_∞	O_∞
U_1	220	7.06E−03	—	2.53E−04	—	7.41E−05	—
	840	1.70E−03	2.12	1.37E−05	4.35	1.11E−06	6.27
	3280	5.12E−04	1.76	8.51E−07	4.08	1.37E−08	6.45
	12960	1.34E−04	1.95	5.72E−08	3.93	1.11E−10	7.01
U_2	220	7.13E−03	—	3.94E−04	—	6.31E−05	—
	840	2.02E−03	1.89	1.92E−05	4.51	9.78E−07	6.22
	3280	6.13E−04	1.75	7.83E−07	4.70	1.46E−08	6.17
	12960	1.36E−04	2.19	5.62E−08	3.84	1.00E−10	7.25
P	100	1.88E−01	—	9.05E−03	—	6.20E−04	—
	400	8.28E−02	1.18	1.08E−03	3.07	1.99E−05	4.97
	1600	4.67E−02	0.83	1.35E−04	3.00	7.29E−07	4.77
	6400	2.45E−02	0.93	1.78E−05	2.92	1.31E−08	5.80

5.2 Fixed-point convergence assessment

This section is dedicated to assess the convergence and efficiency of the fixed-point algorithm. We reuse the previous manufactured solution and consider the three different kinds of fluids, that is, inviscid fluid, viscous fluid with $\nu = 10^{-5}$, and ultra-viscous fluid with $\nu = 1$. Comparisons between centred and upwind approximation for $\Lambda_{k\ell,r}$ are performed and different relaxation factors ($\theta = 1, 0.8, 0.5$) are experimented. The relaxation of the intermediate solutions does not affect the final solution but has a strong impact on the number

of fixed-point iterations required to achieve convergence and, consequently, the execution time of the simulation.

5.2.1 Inviscid fluid

Table 18 gives the number of iterations for the inviscid fluid with successively finer regular triangular Delaunauy meshes and deformed quadrilateral meshes using the $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions.

Table 18 Number fixed-point iterations using regular triangular Delaunay and quadrilateral deformed meshes with $\nu = 0$.

			Delaunay				Deformed			
		θ	440	1544	6296	24840	540	2080	8160	32320
$\mathbb{P}_1$	Centred	1	38	30	33	35	38	30	34	35
		0.8	26	30	25	26	27	31	25	26
		0.5	47	53	43	39	47	53	43	39
	Upwind	1	NC	NC	NC	NC	NC	NC	NC	NC
		0.8	83	114	126	82	83	114	127	82
		0.5	138	55	91	134	138	55	91	132
$\mathbb{P}_3$	Centred	1	90	67	47	33	94	66	40	26
		0.8	26	23	20	22	59	22	20	20
		0.5	47	43	38	39	72	40	38	38
	Upwind	1	NC	NC	NC	NC	NC	NC	NC	NC
		0.8	45	56	93	72	46	60	NC	NC
		0.5	48	47	51	38	47	49	38	38
$\mathbb{P}_5$	Centred	1	NC	147	124	115	82	98	72	47
		0.8	37	30	26	23	31	30	25	33
		0.5	60	52	43	39	55	51	43	39
	Upwind	1	NC	NC	NC	NC	NC	NC	NC	NC
		0.8	37	32	26	26	39	35	30	NC
		0.5	58	52	39	42	49	60	42	42

Convergence. We report that no convergence was achieved in certain cases, in particular when the upwind approximation for $\Lambda_{k\ell,r}$ are used with no relaxation ($\theta = 1$). However, convergence is recovered when considering a smaller relaxation parameter $\theta = 0.8$. On the other hand, we only report one case of no convergence for the centred case with $\theta = 1$.

Efficiency. For the centred case, the minimum number of fixed-point iterations is achieved for $\theta = 0.8$. For the upwind case, the picture is not so clear since a stronger relaxation ($\theta = 0.5$) provides a faster convergence for Delaunay meshes whereas no clear conclusion

can be drawn for the deformed meshes. The number of fixed-point iterations is almost independent of the number of unknowns (DOF) since it is essentially invariant with respect to the successively finer meshes, either for the regular triangular Delaunay meshes or the deformed quadrilateral meshes. We also report the same insensitivity with respect to the polynomial degree for the reconstructions. Finally, we notice that the centred case presents a better convergence than the upwind case. In fact, the centred case achieves convergence even with no relaxation unlike the upwind case and, when both converge, it requires a smaller number of fixed-point iterations. Since the accuracy for the two scenarii is similar, we conclude that the centred approach is more stable, faster and therefore more efficient.

5.2.2 Viscous fluid with $\nu = 10^{-5}$ and $\nu = 1$

In Table 19 we treat the case with $\nu = 10^{-5}$. As in the previous case, the centred approach enjoys a better convergence than the upwind approach and the minimum number of fixed-point iterations is achieved for $\theta = 0.8$.

Table 19 Number fixed-point iterations using regular triangular Delaunay and quadrilateral deformed meshes with $\nu = 10^{-5}$.

			Delaunay				Deformed			
		θ	440	1544	6296	24840	540	2080	8160	32320
$\mathbb{P}_1$	Centred	1	31	33	35	39	208	202	88	73
		0.8	32	42	23	22	30	27	21	20
		0.5	57	69	40	38	45	44	38	37
	Upwind	1	NC	NC	NC	NC	NC	NC	NC	NC
		0.8	84	113	123	81	NC	48	65	NC
		0.5	138	55	91	130	NC	57	49	NC
$\mathbb{P}_3$	Centred	1	102	109	66	38	NC	NC	NC	NC
		0.8	31	28	21	21	61	49	42	28
		0.5	53	49	38	38	100	81	67	46
	Upwind	1	NC	NC	NC	NC	NC	NC	NC	NC
		0.8	45	56	93	72	51	56	42	29
		0.5	48	47	52	39	76	64	63	46
$\mathbb{P}_5$	Centred	1	NC	NC	NC	NC	NC	NC	NC	NC
		0.8	44	32	31	30	56	65	104	32
		0.5	84	56	52	45	93	104	167	53
	Upwind	1	NC	NC	NC	NC	NC	NC	NC	NC
		0.8	37	32	25	25	54	80	60	33
		0.5	57	52	39	42	77	132	95	55

Finally we present in Table 20 the results for $\nu = 1$, where we observe that the two analysed cases are very similar. Such behavior is expected since the convective part of the Navier-Stokes equation is less dominant as the viscosity of the fluid increases. Moreover, the minimum number of fixed-point iterations, achieved with no relaxation, is smaller than the ones for the inviscid case and the viscous case with $\nu = 10^{-5}$. In fact, the nonlinearity contribution of the Navier-Stokes equation turns out to be negligible and as a consequence we mainly deal with the linear Stokes problem.

Table 20 Number fixed-point iterations using regular triangular Delaunay and quadrilateral deformed meshes with $\nu = 1$.

			Delaunay				Deformed			
		θ	440	1544	6296	24840	540	2080	8160	32320
$\mathbb{P}_1$	Centred	1	7	7	7	6	7	7	7	6
		0.8	17	17	17	18	17	17	17	18
		0.5	37	37	37	41	37	37	37	38
	Upwind	1	9	7	7	6	8	7	7	6
		0.8	17	17	17	18	17	17	17	17
		0.5	37	37	37	41	37	37	37	37
$\mathbb{P}_3$	Centred	1	7	7	7	6	7	7	7	7
		0.8	17	17	17	18	17	17	17	17
		0.5	37	37	37	41	37	37	37	37
	Upwind	1	8	7	7	6	8	7	7	6
		0.8	17	17	17	18	17	17	17	17
		0.5	37	37	37	41	37	37	37	37
$\mathbb{P}_5$	Centred	1	7	7	7	6	7	7	7	7
		0.8	17	17	17	18	17	17	17	17
		0.5	37	37	37	41	37	37	37	37
	Upwind	1	8	7	7	6	7	7	7	6
		0.8	17	17	17	18	17	17	17	18
		0.5	37	37	37	41	37	37	37	37

5.3 Time performance assessment

Let us monitor the accuracy as a function of the computational cost of the method. For a given tolerance, we want to determine the mesh and the associated execution time one has to consider to provide a prescribed accuracy for the different types of reconstruction. To do so, we record the execution time, called T_{E}, and we compute the L^2-norm errors of the

velocity and the pressure, given by

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

using successively finer regular triangular Delaunay meshes and deformed quadrilateral meshes. The mesh sizes are the same for the $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions, and have 440, 1544, 6296, and 24840 degrees of freedom for the case of triangular Delaunay meshes and 540, 2080, 8160, and 32320 degrees of freedom for the case of deformed quadrilateral meshes. The simulation were carried out with centred approximations for $\Lambda_{k\ell,r}$ since the upwind case provides comparable execution times.

5.3.1 Inviscid fluid

We plot in Fig. 7 the L^2-norm error as a function of the execution time for $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions using regular triangular Delaunay meshes (left panel) and deformed quadrilateral meshes (right panel) with relaxation factor of $\theta = 0.8$.

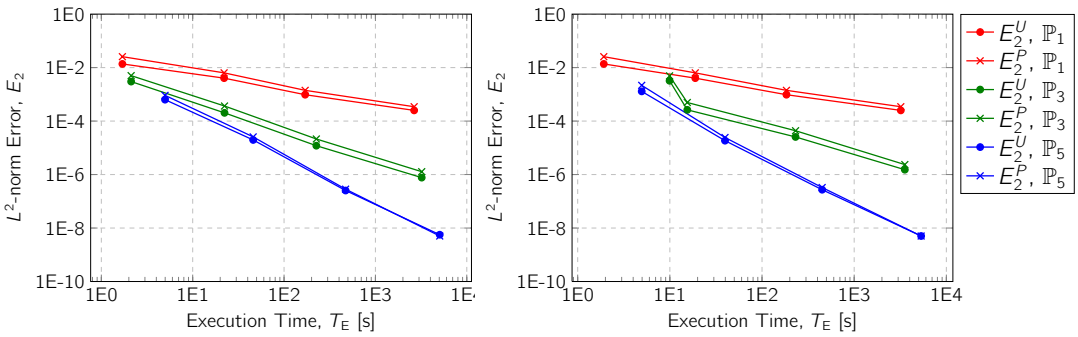

Fig. 7 L^2-norm error as a function of the execution time with $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions and $\theta = 0.8$ using regular triangular Delaunay meshes (left) and deformed quadrilateral meshes (right).

Clearly the $\mathbb{P}_5$ reconstruction has the lowest computational cost to reach a given error. As an example, to provide a numerical solution such that the L^2-error of the velocity is lower than 10^{-6}, the $\mathbb{P}_5$ version requires around 120s with a mesh of 1000 cells (roughly

speaking), the $\mathbb{P}_3$ reconstruction needs 1200s with a mesh of 6000 cells, and the $\mathbb{P}_1$ second-order method performs the computation in 10^8s (about 3 years) with a mesh of about 10^7 cells (estimated value).

We now repeat the tests but with $\theta = 0.5$. Taking in consideration the results presented in Fig. 8, we conclude again that the $\mathbb{P}_5$ reconstruction is more efficient for a given tolerance when compared to the $\mathbb{P}_3$ which is also more efficient compared to the $\mathbb{P}_1$ reconstruction.

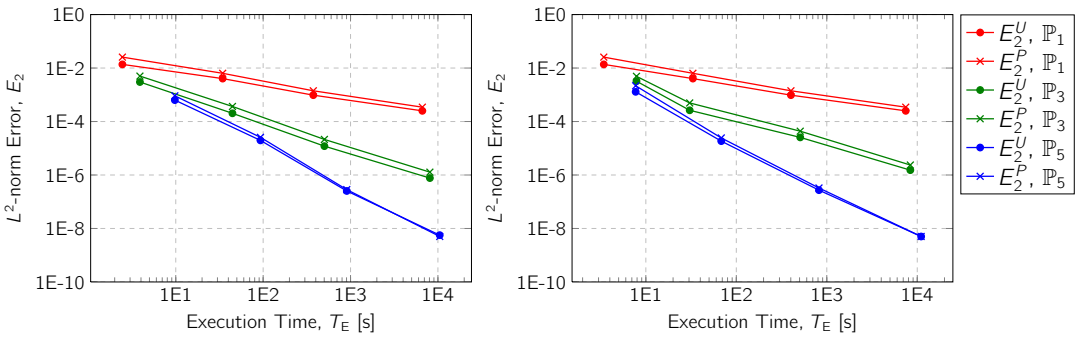

Fig. 8 L^2-norm error as a function of the execution time with $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions and $\theta = 0.5$ using regular triangular Delaunay meshes (left) and deformed quadrilateral meshes (right).

5.3.2 Viscous fluid with $\nu = 10^{-5}$

We now deal with the viscous fluid with $\nu = 10^{-5}$ and plot in Fig. 9 and in Fig. 10 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 regular triangular Delaunay meshes (left panel) and deformed quadrilateral meshes (right panel) with relaxation factor of $\theta = 0.8$ and $\theta = 0.5$, respectively. When using regular triangular Delaunay meshes, the $\mathbb{P}_5$ reconstruction is clearly the most efficient followed by the $\mathbb{P}_3$ reconstruction. For deformed quadrilateral meshes the $\mathbb{P}_3$ reconstruction and the $\mathbb{P}_5$ reconstruction seems to be comparable for coarse and middle range meshes. Indeed, we have noticed in Table 19 that the $\mathbb{P}_5$ requires a few more iterations than $\mathbb{P}_3$ to achieve convergence which leads to a higher execution time.

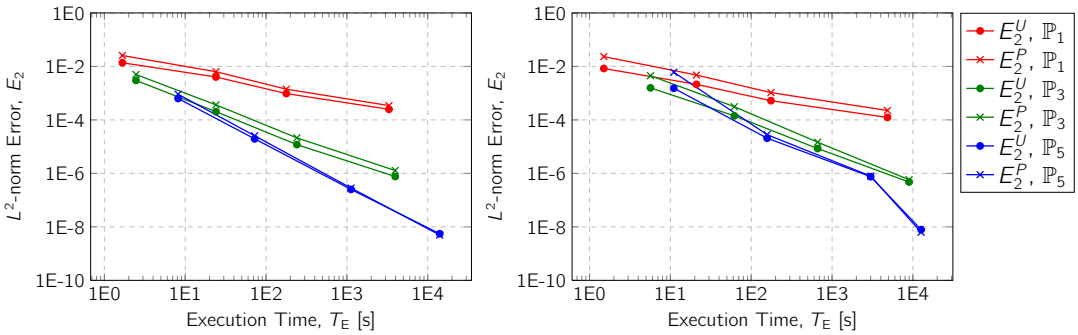

Fig. 9 L^2-norm error as a function of the execution time with $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions and $\theta = 0.8$ using regular triangular Delaunay meshes (left) and deformed quadrilateral meshes (right).

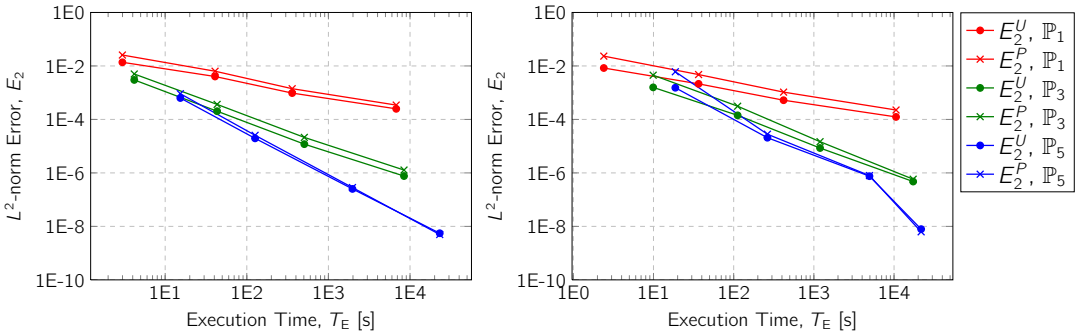

Fig. 10 L^2-norm error as a function of the execution time with $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions and $\theta = 0.5$ using regular triangular Delaunay meshes (left) and deformed quadrilateral meshes (right).

5.3.3 Viscous fluid with $\nu = 1$

Finally, we deal with the ultra-viscous fluid ($\nu = 1$) and plot in Fig.s 11, 12, and 13 the L^2-norm error as a function of the execution time for regular triangular Delaunay meshes (left panel) and deformed quadrilateral meshes (right panel) with relaxation factor $\theta = 1$, $\theta = 0.8$, and $\theta = 0.5$, respectively. We first notice that the lines corresponding to the velocity and the pressure errors do not overlap anymore as in the previous cases since the pressure convergence is one order lower compared to the velocity. Secondly, the $\mathbb{P}_5$ reconstruction is clearly more efficient followed by the $\mathbb{P}_3$ reconstruction which justifies the interest of very-high order schemes.

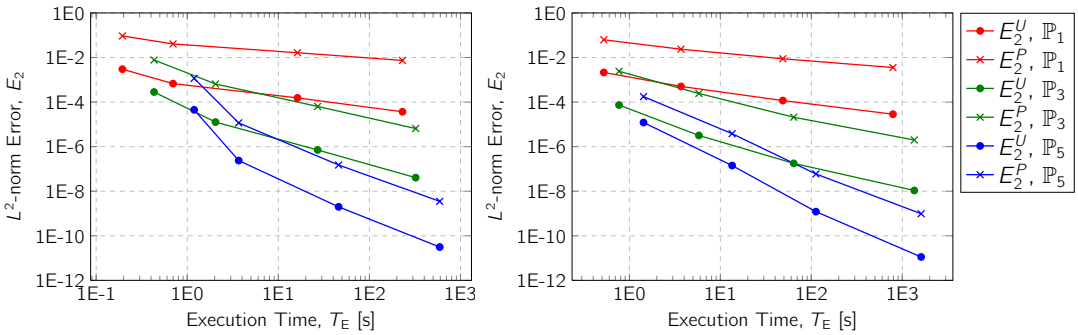

Fig. 11 L^2-norm error as a function of the execution time with $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions and $\theta = 1$ using regular triangular Delaunay meshes (left) and deformed quadrilateral meshes (right).

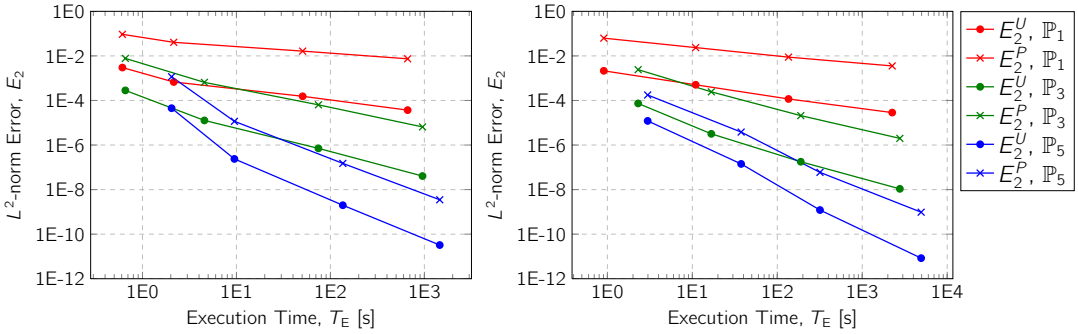

Fig. 12 L^2-norm error as a function of the execution time with $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions and $\theta = 0.8$ using regular triangular Delaunay meshes (left) and deformed quadrilateral meshes (right).

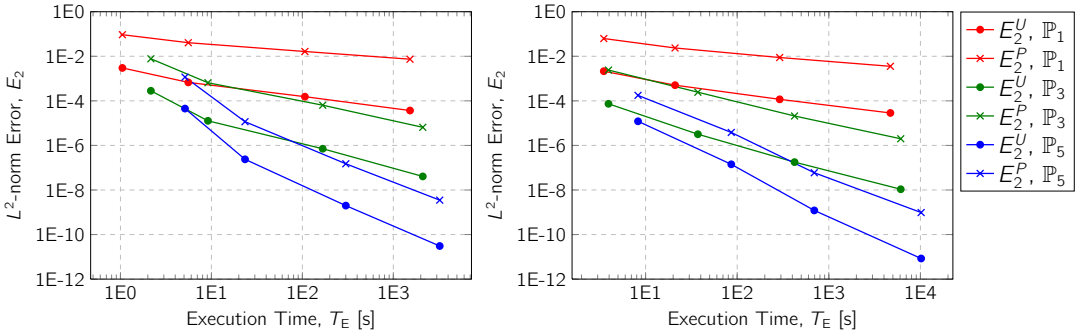

Fig. 13 L^2-norm error as a function of the execution time with $\mathbb{P}_1$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions and $\theta = 0.5$ using regular triangular Delaunay meshes (left) and deformed quadrilateral meshes (right).

6 Lid-driven square cavity problem

The 2D steady-state lid-driven square cavity problem is a popular benchmark to validate numerical methods in solving the Navier-Stokes equations. A unit square cavity $\Omega =]0,1[^2$ containing a viscous fluid with $U_{\mathrm{D}}(x) = (1,0)$, $x \in \partial\Omega$, $x_2 = 1$, $U_{\mathrm{D}}(x) = (0,0)$, $x \in \partial\Omega$, $x_2 \neq 1$ (static walls). In order to deal with one parameter model (since $\nu > 0$ we can compute the Reynolds number), we consider the dimensionless variables $L^* = L/L_{\mathrm{ref}}$ for the length, $U^* = U/U_{\mathrm{ref}}$ for the flow velocity, $P^* = PL_{\mathrm{ref}}/\mu U_{\mathrm{ref}}$ for the pressure where L_{ref} and U_{ref} are reference values for the length and the magnitude of the flow velocity, respectively, and μ is the dynamic viscosity of the fluid. Under this rescaling, the non-dimensionalized steady-state incompressible Navier-Stokes problem is given by

$$\begin{aligned} \nabla \cdot (Re\, U^* \otimes U^* - \nabla U^* + P^* I) &= 0, && \text{in } \Omega, \\ \nabla \cdot U^* &= 0, && \text{in } \Omega, \end{aligned}$$

where Re is the Reynolds number given by $Re = \rho U_{\mathrm{ref}} L_{\mathrm{ref}}/\mu$. The linearization of the non-dimensionalized Navier-Stokes problem yields

$$\begin{aligned} \nabla \cdot (Re\, U^* \otimes \Lambda^* - \nabla U^* + P^* I) &= 0, && \text{in } \Omega, \\ \nabla \cdot U^* &= 0, && \text{in } \Omega, \end{aligned}$$

where Λ^* stands for a guess of U^*. We solve this system of equations as before and having the velocity in hand, we compute the vorticity of the flow, $\omega \equiv \omega(x) = \nabla \times U^*$. The stream function, $\psi \equiv \psi(x)$, for the lid-driven cavity problem can be found from the vorticity through the Poisson equation $\nabla^2 \psi = -\omega$ in Ω, with the Dirichlet boundary condition $\psi_{\mathrm{D}}(x) = 0$ on $\partial\Omega$.

We consider two cases corresponding to Reynolds numbers $Re = 100$ and $Re = 400$. The simulations employ structured quadrilateral meshes with 10×10, 20×20, 40×40, and 80×80 cells using a centred approximations for $\Lambda_{k\ell,r}$ and $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions. For all the cases the relaxation coefficient was set $\theta = 0.5$.

Figure 14 displays the streamlines for $Re = 100$. Clearly the $\mathbb{P}_5$ reconstruction systematically better captures the stream function contours compared to the other reconstructions using the same mesh, followed by the $\mathbb{P}_3$ and the $\mathbb{P}_2$ reconstructions, whereas the $\mathbb{P}_1$ reconstruction always provides the worst results. The $\mathbb{P}_5$ reconstruction manages to capture (even if not very accurately) the secondary vortices with a mesh of 20×20 cells whereas the $\mathbb{P}_3$ reconstruction requires 20×20 cells mesh to capture the right secondary vortex and a 40×40 cells mesh to capture the left secondary vortex. On the other hand, the $\mathbb{P}_2$ reconstruction needs the 40×40 cells mesh to catch the right secondary vortex whereas the 80×80 cells mesh is necessary to distinguish the left secondary vortex. For $\mathbb{P}_1$ reconstruction where only 80×80 cells mesh enables to capture both vortices. To achieve the same quality for the secondary vortices captured by the $\mathbb{P}_5$ reconstruction with a mesh of 40×40 cells, a 80×80 cells mesh with the $\mathbb{P}_3$ reconstruction is required. The $\mathbb{P}_1$ reconstruction never achieves such a quality.

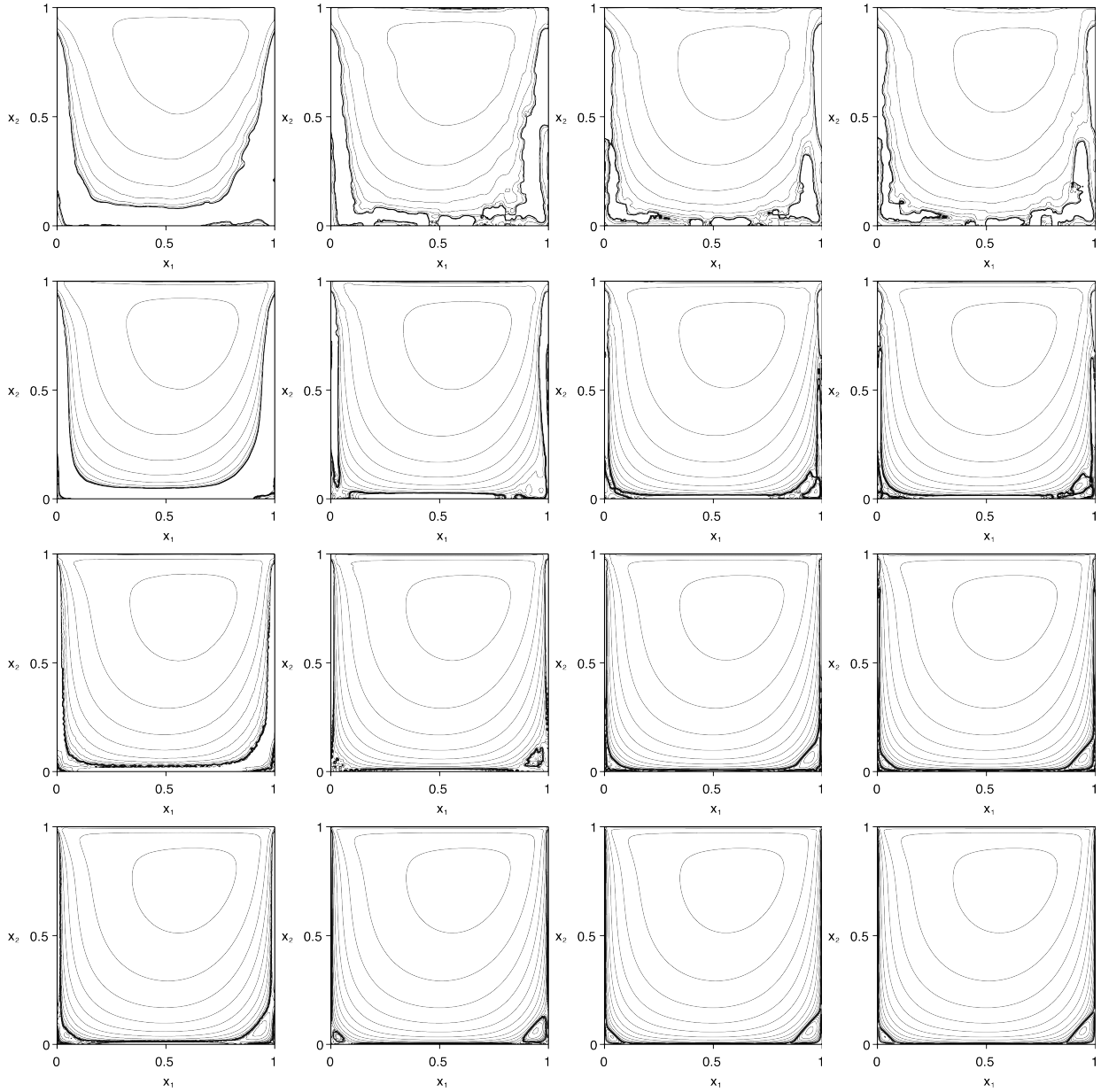

Fig. 14 Streamlines for the lid-driven cavity problem with $Re = 100$ and $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions from the first to the last rows using meshes of 10×10 cells, 20×20 cells, 40×40 cells, and 80×80 cells from the first to the last columns.

Tables 21 and 22 provide the coordinates of the center of the primary vortex and of the centers of the secondary vortices, respectively, as well as the associated values of ψ function

for each simulation carried out with $Re = 100$. We also compare the locations of the vortices obtained by our simulation with the values obtained in the literature [39–43].

The simulation carried out with the $\mathbb{P}_5$ reconstruction situates the centre of the primary vortex at $x = (0.6141, 0.7359)$ with $\psi(x) = -0.104$ which matches well with the values obtained from the literature. Notice that the location does not change for the 40×40 and 80×80 cells meshes and the same invariance holds with the $\mathbb{P}_3$ reconstruction. Suck vortex center is achieved by $\mathbb{P}_2$ reconstruction only with the 40×40 cells mesh whereas we need a 80×80 cells mesh to obtain the same accuracy using the $\mathbb{P}_1$ reconstruction. Concerning the secondary vortices, we observe a large deviation of the vortices centre between the different authors. Nonetheless, the values we have obtained both for the left and for the right vortices converge when the mesh is refined.

Table 21 Strength and location of the center of the primary vortex for the lid-driven square cavity problem with $Re = 100$.

	Mesh	$\psi_{\min}$	x_1	x_2
$\mathbb{P}_1$	10×10	-0.112	0.6047	0.7797
	20×20	-0.109	0.6047	0.7609
	40×40	-0.105	0.6078	0.7422
	80×80	-0.104	0.6172	0.7359
$\mathbb{P}_2$	10×10	-0.118	0.5922	0.7234
	20×20	-0.106	0.6141	0.7359
	40×40	-0.104	0.6109	0.7359
	80×80	-0.104	0.6172	0.7391
$\mathbb{P}_3$	10×10	-0.111	0.6234	0.7547
	20×20	-0.105	0.6172	0.7391
	40×40	-0.104	0.6141	0.7359
	80×80	-0.104	0.6141	0.7359
$\mathbb{P}_5$	10×10	-0.103	0.6109	0.7547
	20×20	-0.103	0.6109	0.7391
	40×40	-0.104	0.6141	0.7359
	80×80	-0.104	0.6141	0.7359
[43]	40×40	-0.103	0.6125	0.7375
[40]	128×128	-0.103	0.6172	0.7344
[42]	255×255	-0.103	0.6196	0.7373

Table 22 Strength and location of the centers of the secondary vortices for the lid-driven square cavity problem with $Re = 100$.

		Bottom left			Bottom right		
	Mesh	$\psi_{\max}$	x_1	x_2	$\psi_{\max}$	x_1	x_2
$\mathbb{P}_1$	10×10	NF	—	—	NF	—	—
	20×20	NF	—	—	NF	—	—
	40×40	3.688E−5	0.0297	0.0859	7.341E−5	0.9453	0.1047
	80×80	8.460E−6	0.0422	0.0391	2.830E−5	0.9359	0.0672
$\mathbb{P}_2$	10×10	NF	—	—	NF	—	—
	20×20	NF	—	—	NF	—	—
	40×40	NF	—	—	1.200E−5	0.9391	0.0609
	80×80	1.567E−6	0.0297	0.0391	1.253E−5	0.9453	0.0641
$\mathbb{P}_3$	10×10	NF	—	—	8.382E−4	0.9422	0.0703
	20×20	NF	—	—	2.022E−5	0.9109	0.0453
	40×40	4.826E−6	0.0203	0.0578	1.505E−5	0.9422	0.0641
	80×80	2.036E−6	0.0359	0.0328	1.301E−5	0.9422	0.0609
$\mathbb{P}_5$	10×10	NF	—	—	9.160E−4	0.9422	0.0734
	20×20	7.590E−6	0.0672	0.0297	2.351E−5	0.9234	0.0640
	40×40	2.838E−6	0.0391	0.0359	1.388E−5	0.9422	0.0609
	80×80	1.952E−6	0.0328	0.0359	1.292E−5	0.9522	0.0609
[43]	40×40	1.83E−6	0.0375	0.0375	1.45E−5	0.9375	0.0625
[40]	128×128	1.75E−6	0.0313	0.0391	1.25E−5	0.9453	0.0625
[42]	255×255	1.72E−6	0.0392	0.0353	1.22E−5	0.9451	0.0627

We now consider $Re = 400$, ploting in Fig. 15 the streamlines. We achieve the same conclusions for the $Re = 400$ case as for the $Re = 100$ one, that is, the $\mathbb{P}_5$ reconstruction always capture better contours for the stream function compared to the other reconstructions using the same mesh, followed by the $\mathbb{P}_3$ and the $\mathbb{P}_2$ reconstructions, where the $\mathbb{P}_1$ reconstruction is always the worst case. Similarly, the $\mathbb{P}_5$ and $\mathbb{P}_3$ reconstructions can capture (even if not very accurately) both the secondary vortices with a mesh of 20×20 cells. On the other hand, the $\mathbb{P}_2$ reconstruction needs a mesh of 40×40 cells to capture the secondary vortices while the $\mathbb{P}_1$ reconstruction requires a mesh of 80×80 cells. Using the 40×40 cells mesh, the $\mathbb{P}_3$ and $\mathbb{P}_5$ reconstruction manage to provide comparable contours for the stream function while the $\mathbb{P}_2$ reconstruction needs a mesh of 80×80 cells for the same quality. On the other hand, the $\mathbb{P}_1$ reconstruction does not provide comparable quality even with the 80×80 cells mesh.

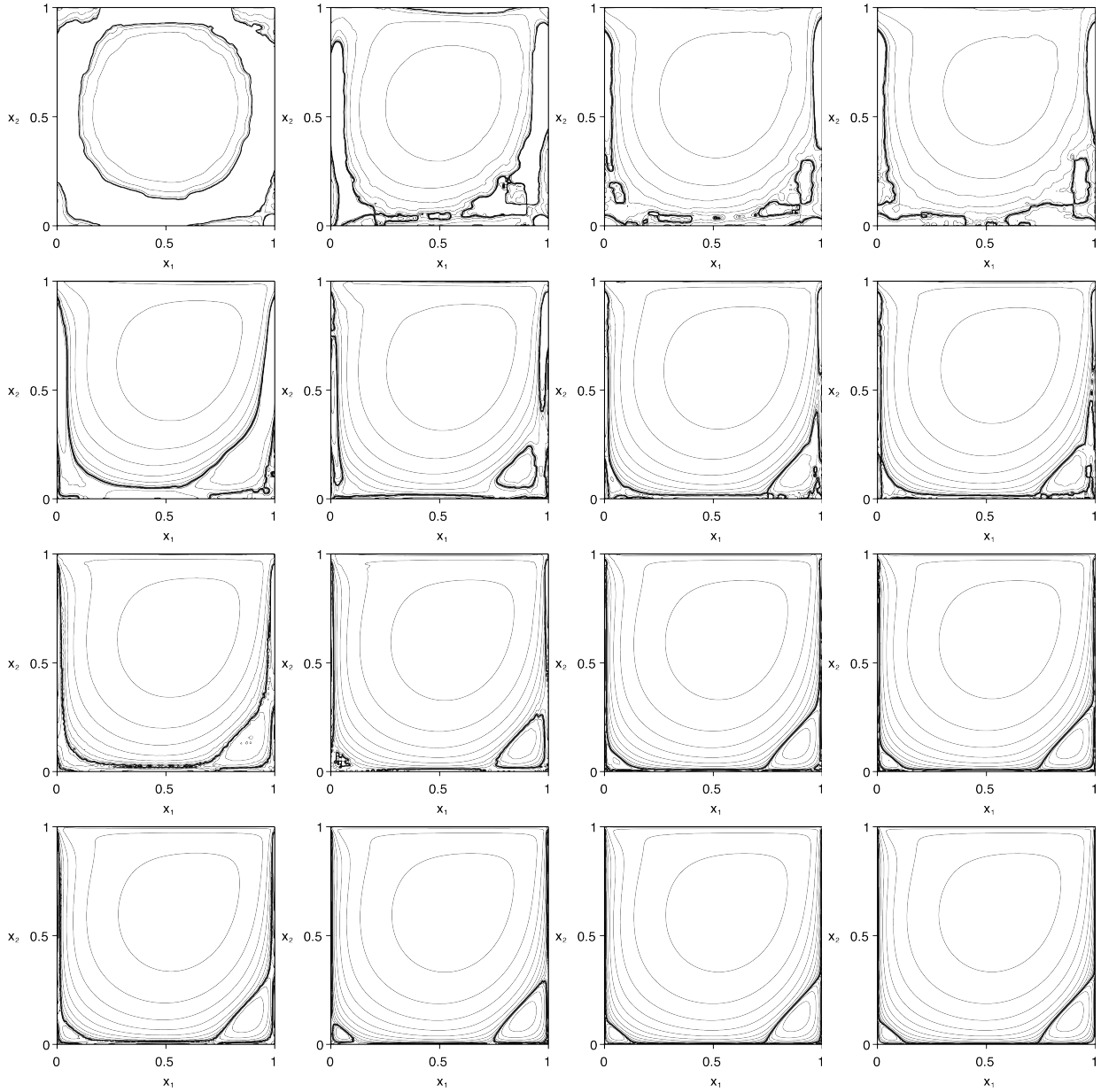

Fig. 15 Streamlines for the lid-driven cavity problem with $Re = 400$ and $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_3$, and $\mathbb{P}_5$ polynomial reconstructions from the first to the last rows using meshes of 10×10 cells, 20×20 cells, 40×40 cells, and 80×80 cells from the first to the last columns.

Tables 23 and 24 provide the coordinates of the center of the primary vortex and of the centers of the secondary vortices, respectively, as well as the associated values of ψ function

for each simulation carried out with $Re = 400$. The values obtained in the works [39–43] are also reported for comparison purposes.

The $\mathbb{P}_5$ reconstruction provides the location of a primary vortex at $x = (0.5547, 0.6047)$ with $\psi(x) = -0.114$ which has a good match with the values obtained from the literature. Such a result is also achieved with the $\mathbb{P}_2$ and $\mathbb{P}_3$ reconstructions whereas the $\mathbb{P}_1$ fails to provide the correct location. A good match both for the left and right vortices is obtained when compared to the values from the literature.

Table 23 Strength and location of the center of the primary vortex for the lid-driven square cavity problem with $Re = 400$.

	Mesh	$\psi_{\min}$	x_1	x_2
$\mathbb{P}_1$	10×10	NC	—	—
	20×20	-0.119	0.5641	0.6297
	40×40	-0.116	0.5547	0.6391
	80×80	-0.114	0.5547	0.6078
$\mathbb{P}_2$	10×10	NC	—	—
	20×20	-0.125	0.5453	0.6116
	40×40	-0.116	0.5516	0.6047
	80×80	-0.114	0.5547	0.6047
$\mathbb{P}_3$	10×10	-0.129	0.5484	0.6047
	20×20	-0.118	0.5516	0.6016
	40×40	-0.115	0.5547	0.6047
	80×80	-0.114	0.5547	0.6047
$\mathbb{P}_5$	10×10	-0.109	0.5547	0.6203
	20×20	-0.112	0.5547	0.6109
	40×40	-0.114	0.5547	0.6078
	80×80	-0.114	0.5547	0.6047
[43]	80×80	-0.113	0.5500	0.6125
[40]	128×128	-0.114	0.5547	0.6055
[42]	255×255	-0.112	0.5608	0.6078

Table 24 Strength and location of the centers of the secondary vortices for the lid-driven square cavity problem with $Re = 400$.

		Bottom left			Bottom right		
	Mesh	$\psi_{\max}$	x_1	x_2	$\psi_{\max}$	x_1	x_2
$\mathbb{P}_1$	10×10	NC	—	—	NC	—	—
	20×20	NF	—	—	2.326E−3	0.8516	0.1484
	40×40	8.093E−5	0.0828	0.0578	1.024E−3	0.8891	0.1328
	80×80	3.035E−5	0.0672	0.0422	7.585E−4	0.8734	0.1266
$\mathbb{P}_2$	10×10	NC	—	—	NC	—	—
	20×20	NF	—	—	8.762E−4	0.8641	0.1328
	40×40	1.641E−5	0.0547	0.0547	6.763E−4	0.8891	0.1297
	80×80	1.419E−5	0.0516	0.0516	6.484E−4	0.8859	0.1266
$\mathbb{P}_3$	10×10	NF	—	—	NF	—	—
	20×20	4.958E−5	0.0359	0.0922	7.357E−4	0.9016	0.1359
	40×40	1.967E−5	0.0516	0.0484	6.580E−4	0.8859	0.1234
	80×80	1.512E−5	0.0516	0.0484	6.447E−4	0.8859	0.1234
$\mathbb{P}_5$	10×10	NF	—	—	NF	—	—
	20×20	4.076E−5	0.0547	0.0609	6.465E−4	0.8703	0.1266
	40×40	1.695E−5	0.0516	0.0516	6.391E−4	0.8859	0.1234
	80×80	1.476E−5	0.0516	0.0484	6.427E−4	0.8859	0.1234
[43]	80×80	1.30E−5	0.0500	0.0500	6.48E−4	0.8875	0.1250
[40]	128×128	1.42E−5	0.0508	0.0469	6.42E−4	0.8906	0.1250
[42]	255×255	1.30E−5	0.0549	0.0510	6.19E−4	0.8902	0.1255

7 Conclusions

We have presented a high-order accurate finite volume scheme to solve the bidimensional incompressible Navier-Stokes and Euler problems based on a new class of polynomial reconstructions. The scheme achieves an effective sixth-order accuracy for the velocity and a sixth- or fifth-order accuracy for the pressure depending on the Reynolds number associated with the flow. Detailed numerical tests were carried out to prove the efficiency and the stability of the new high-order scheme. We show that the use of very high-order finite volume schemes decreases the computational effort required to achieve a given accurate solutions. We have also carried out the simulation of the classical lid-driven square cavity problem and we have numerically proved the capacity of the high-order scheme to retrieve accurate solutions for the stream function with coarse meshes.

Acknowledgements This research was financed by the International Centre for Mathematics and Computer Science in Toulouse – CIMI, 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. Y. Adam, Highly accurate compact implicit methods and boundary conditions, J. Comput. Phys. 24 (1977) 10–22.
2. T.J. Barth, Recent developments in high order *k*-exact reconstruction on unstructured meshes, AIAA Paper 93-0668, 1993.
3. T.J. Barth, P.O. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper 90–0013, 1990.
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. D. L. Brown, R. Cortez, M. L. Minion, Accurate Projection Methods for the Incompressible NavierStokes Equations, Journal of Computational Physics 168, (2001) 464–499.
8. S. Clain, G.J. Machado, A very high-order finite volume method for the time-dependent convectiondiffusion problem with Butcher tableau extension, Computers and Mathematics with Applications 68 (2014) 1292–1311.
9. R. Costa, S. Clain, G.J. Machado, A sixth-order finite volume scheme for the steady-state incompressible Stokes equations on staggered unstructured meshes, submitted, available at https://hal.archives-ouvertes.fr/hal-01107313.
10. L. Cueto-Felgueroso, I. Colominas, X. Nogueira, F. Navarrina, M. Casteleiro, Finite volume solvers and Moving Least-Squares approximations for the compressible Navier-Stokes equations on unstructured grids, Computer Methods in Applied Mechanics and Engineering, 196 (2007) 4712–4736.
11. 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.

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, J.C. Latché, R. Herbin, B. Piar, Convergence of a locally stabilized collocated finite volume scheme for incompressible flows, M2AN 43 (2009) 889–927.

15. J.H. Ferziger, M. Perić, Computational methods for fluids dynamics, Springer-Verlag, Berlin, 1996.

16. A. Fosso, H. Deniau, F. Sicot, P. Saguat, Curvilinear finite volume schemes using high-order compact interpolation, J. Comput. Phys., 229 (2010) 5090–5122.

17. 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.

18. 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.

19. R.-S. Hirsh, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, J. Comput. Phys. 19 (1975) 90–109.

20. 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.

21. L. Ivan, C.P.T. Groth, High-order solution-adaptative central essentially non-oscillatory (CENO) method for viscous flows, AIAA Paper 2011-367, 2011.

22. N.A. Kampanis, J.A. Ekaterinaris, A staggered grid, high-order accurate method for the incompressible Navier-Stokes Equations, J. Comput. Phys. 215 (2006) 589–613.

23. J.W. Kim, D.J. Lee, Optimized compact finite difference schemes with maximum resolution, AIAA Journal 34 (5) (1996) 887–893.

24. M.H. Kobayashi, On a class of Padé finite volume methods, Journal of Computational Physics 156 (1999) 137–180.

25. P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods, Math Comput 87 (1981) 141–158.

26. S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.

27. L. R. Palacios, High-order finite volume methods based on moving least squares for computational fluid dynamics. Application to all-speed and incompressible flows on unstructured grids, PhD Thesis, Universidade da Coruña (2015).

28. 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.

29. C. Michalak, C. Ollivier-Gooch, Unstructured high-order accurate finite-volume solutions of the Navier-Stokes equations, AIAA Paper 2009-954, 2009.

30. S.V. Patankar, Numerical heat transfer and fluid flow, Hemisphere, New-York, 1980.

31. J.M.C. Pereira, M.H. Kobayashi, J.C.F. Pereira, A fourth-order-accurate Finite volume Compact method for the incompressible Navier-Stokes solutions, J. Comput. Phys. 167 (2001) 217–243.

32. M. Piller, E. Stalio, Finite volume compact schemes on staggered grids, J. Comput. Phys. 197 (2004) 299–340.

33. 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.

34. 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.

35. S. Shang, X. Zhao, S. Bayyuk, Generalized formulations for the Rhie-Chow interpolation, J. Comput. Phys. 258 (2014) 880–914.

36. R. Temam, Navier-Stokes equations, Theory and numerical analysis, North-Holland, Amsterdam, 1987.

37. 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.

38. O.C. Zienkiewicz, R.L. Taylor, P. Nithiarasu, The finite element method for fluid dynamics, Butterworth-Heinemann, Waltham, 2014.

39. S.P. Vanka, Block-implicit multigrid solution of Navier-Stokes equations in primitive variables, J. Comput. Phys. 65 (1986) 138–158.

40. U. Ghia, K.N. Ghia, C.T. Shin, High Re-solution for incompressible Navier-Stokes equation and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.

41. R. Schreiber, H.B. Keller, Driven cavity flows by efficient numerical techniques, J. Comput. Phys. 49 (1983) 310–333.

42. S. Hou, Q. Zou, S. Chen, G. Doolen, A. Cogley, Simulation of cavity flows by the lattice Boltzmann method, J. Comput. Phys. 118 (1995) 329–347.

43. M.M. Gupta, J.C, Kalita, A new paradigm for solving Navier-Stokes equations: streamfunctionvelocity formulation, J. Comput. Phys. 207 (2005) 52–68.