MOX-Report No. 47/2021

An efficient and accurate implicit DG solver for the incompressible Navier-Stokes equations

Orlando, G; Della Rocca, A; Barbante, P. F.; Bonaventura, L.;

Parolini, N.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it http://mox.polimi.it

An efficient and accurate implicit DG solver for the incompressible Navier-Stokes equations

Giuseppe Orlando[(1)], Alessandro Della Rocca[(2)]
Paolo Francesco Barbante[(1)], Luca Bonaventura[(1)], Nicola Parolini[(1)]

July 16, 2021

[(1)] MOX - Dipartimento di Matematica, Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
giuseppe.orlando@polimi.it, paolo.barbante@polimi.it,
luca.bonaventura@polimi.it, nicola.parolini@polimi.it

[(2)] Tenova S.p.A.
Global R&D
Via Albareto 31, 16153 Genova, Italy
alessandro.dellarocca@tenova.com

Keywords: Navier-Stokes equations, incompressible flows, Discontinuous Galerkin methods, implicit methods, ESDIRK methods.

AMS Subject Classification: 65M08, 65N08, 65N12, 65Z05, 76R50

Abstract

We propose an efficient, accurate and robust implicit solver for the incompressible Navier-Stokes equations, based on a DG spatial discretization and on the TR-BDF2 method for time discretization. The effectiveness of the method is demonstrated in a number of classical benchmarks, which highlight its superior efficiency with respect to other widely used implicit approaches. The parallel implementation of the proposed method in the framework of the *deal.II* software package allows for accurate and efficient adaptive simulations in complex geometries, which makes the proposed solver attractive for large scale industrial applications.

1 Introduction

The efficient numerical solution of the incompressible Navier-Stokes equations is one of the most relevant goals of computational fluid dynamics. A great number of methods have been proposed in the literature, see for example, among many others, the reviews in [43, 44]. Since the seminal proposals [19, 52], projection methods [32] have become very popular for the time discretization of this problem. Several spatial discretization approaches have been proposed and finite volume techniques using unstructured meshes [24] have become the state of the art for industrial applications, in particular when implemented in parallel software packages like OpenFoam [17, 36, 56]. Indeed, in previous work by one of the authors [47], a wide range of projection methods was implemented in OpenFoam and their performance was compared, as a preliminary step towards the development of a computational fluid dynamics tool for combustion simulations of industrial interest. On the other hand, high order finite elements, both in their continuous and discontinuous versions [29, 38], have gained increasing popularity in the academic community and also in many applications, but are still far from being the reference tool for industrial use. More specifically, Discontinuous Galerkin methods for the Navier-Stokes equations have been proposed by many authors, we refer for example to [8, 21, 22, 23, 28, 49].

In this work, we seek to combine, on the one hand, accurate and flexible discontinuous finite element spatial discretizations, and on the other hand, efficient and unconditionally stable time discretizations, following an approach that has been shown to be quite successful for applications to numerical weather prediction in [54, 55]. Building on the experience of [47], we propose an accurate, efficient and robust projection method, based on the second order TR-BDF2 method [6, 35, 54]. This solver is implemented using discontinuous finite elements, in the framework of the numerical library *deal.II* [5], in order to provide a reliable and easily accessible tool for large scale industrial applications. It is important to remark that time discretizations of the Navier-Stokes equations based on accurate implicit solvers have been proposed in a number of papers, see among many others [7, 9, 46, 50, 51]. The specific combination of techniques presented in this work does not entail major conceptual novelties with respect to any of the above references, but we claim that it constitutes an optimal combination for the development of a second order $h-$adaptive flow solver that can be competitive for industrial applications with more conventional finite volume techniques. Furthermore, while the TR-BDF2 method is only second order in time, the wide range of simulations presented in [54] show that this method still allows to achieve quite accurate results even when coupled to higher order discretizations in space. The paper is organized as follows: the time discretization approach is outlined and discussed in Section 2. The

spatial discretization is presented in Section 3. Some implementation issues, the validation of the proposed method and its application to a number of significant CFD benchmarks are reported in Section 4. Some conclusions and perspectives for future work are described in Section 5.

2 The Navier-Stokes equations and the time discretization strategy

Let $\Omega \subset \mathbb{R}^d, 2 \leq d \leq 3$ be a connected open bounded set with a sufficiently smooth boundary $\partial\Omega$ and denote by $\mathbf{x}$ the spatial coordinates and by t the temporal coordinate. We consider the classical unsteady incompressible Navier-Stokes equations, written in non-dimensional form as:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{u}\otimes\mathbf{u}) + \nabla p &= \frac{1}{Re}\Delta\mathbf{u} + \mathbf{f} \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned} \tag{1}$$

for $\mathbf{x} \in \Omega$, $t \in (0, T]$, supplied with suitable initial and boundary conditions. Here T is the final time, $\mathbf{u}$ is the fluid velocity, p is the pressure and Re is the Reynolds number, which is usually defined as $Re = UL/\nu$, where U denotes a reference value of the velocity magnitude, L a reference length scale and ν the fluid kinematic viscosity. The velocity $\mathbf{u}$ and the pressure p are coupled together by the incompressibility constraint in (1), which leads, after space discretization, to a system of differential and algebraic equations whose numerical solution presents several difficulties widely discussed in the literature. Furthermore, in the specific case of projection methods, difficulties arise in choosing the boundary conditions to be imposed for the Poisson equation which is to be solved at each time step to compute the pressure, see e.g. the discussion in [32].

An alternative that allows to avoid or reduce some of these problems is the so-called artificial compressibility formulation, originally introduced in [18]. In this formulation, the incompressibility constraint is relaxed and a time evolution equation for the pressure is introduced, which is characterized by an artificial sound speed c, so as to obtain

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{u}\otimes\mathbf{u}) + \nabla p &= \frac{1}{Re}\Delta\mathbf{u} + \mathbf{f} \\ \frac{1}{c^2}\frac{\partial p}{\partial t} + \nabla\cdot\mathbf{u} &= 0. \end{aligned} \tag{2}$$

For the sake of simplicity, we shall only consider $\mathbf{f} = \mathbf{0}$ and Dirichlet boundary conditions for the velocity, i.e., $\mathbf{u}|_{\partial\Omega} = \mathbf{u}_D(t)$, while we consider homogeneous Neumann boundary conditions for the pressure. While most

commonly discretized by explicit methods, see e.g. [41, 48] among many others, implicit methods have also been applied to this formulation, see e.g. [20, 40, 45].

Our goal here is to extend the projection method based on the TR-BDF2 scheme introduced in [47] for the formulation (1) to the time discretization of system (2). This allows to avoid the introduction of stabilization parameters and to exploit the special properties of the TR-BDF2 method, which will be reviewed here briefly. Introducing a discrete time step $\Delta t = T/N$ and discrete time levels $t^n = n\Delta t$, $n = 0, \dots, N$, for a generic time dependent problem $\boldsymbol{u}' = \mathcal{N}(\boldsymbol{u})$ the incremental form of the TR-BDF2 method can be described in terms of two stages, the first from t^n to $t^{n+\gamma} = t^n + \gamma\Delta t$ and the second from $t^{n+\gamma}$ to t^{n+1}, which can be written as:

$$\begin{aligned} \frac{\boldsymbol{u}^{n+\gamma} - \boldsymbol{u}^n}{\gamma \Delta t} &= \frac{1}{2}\mathcal{N}\left(\boldsymbol{u}^{n+\gamma}\right) + \frac{1}{2}\mathcal{N}\left(\boldsymbol{u}^n\right) \\ \frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^{n+\gamma}}{(1-\gamma)\,\Delta t} &= \frac{1}{2-\gamma}\mathcal{N}\left(\boldsymbol{u}^{n+1}\right) + \frac{1-\gamma}{2\,(2-\gamma)}\mathcal{N}\left(\boldsymbol{u}^{n+\gamma}\right) + \frac{1-\gamma}{2\,(2-\gamma)}\mathcal{N}\left(\boldsymbol{u}^n\right). \end{aligned} \tag{3}$$

Here, $\boldsymbol{u}^n$ denotes the approximation at time $n = 0, ..., N$. Notice that, in order to guarantee L-stability, one has to choose $\gamma = 2 - \sqrt{2}$. This second order implicit method, originally introduced in [6] as a combination of the Trapezoidal Rule (or Crank-Nicolson) method and of the Backward Differentiation Formula method of order 2, has been fully analyzed in [35]. While we will use here its original formulation, the method was shown in [35] to be an L-stable Explicit first step, Diagonally Implicit Runge Kutta method (ESDIRK). Explicit methods that complement TR-BDF2 as second order IMEX pairs have been introduced in [30] and succesfully employed in [12], [26]. Unconditionally strong stability preserving extensions of TR-BDF2 have been derived in [14]. While the third order method that constitutes an embedded pair with TR-BDF2 is only conditionally stable, see the discussion in [35], a first order embedded method is derived in [39], thus allowing for efficient time adaptation strategies. Finally, the analysis presented in [13] shows that the method is optimal among second order methods for typical structural mechanics equations, thus making it an excellent candidate also for applications to fluid-structure interaction problems. While we do not pursue these developments in the present work, we would like to highlight these features as strong motivations for our specific choice of the time discretization method.

Following then the projection approach described in [47] and applying method (3) to system (2), the momentum predictor equation for the first stage reads:

$$\begin{aligned}
&\frac{\mathbf{u}^{n+\gamma,*}-\mathbf{u}^{n}}{\gamma\Delta t}-\frac{1}{2Re}\Delta\mathbf{u}^{n+\gamma,*}+\frac{1}{2}\nabla\cdot\left(\mathbf{u}^{n+\gamma,*}\otimes\mathbf{u}^{n+\frac{\gamma}{2}}\right)=\\
&\frac{1}{2Re}\Delta\mathbf{u}^{n}-\frac{1}{2}\nabla\cdot\left(\mathbf{u}^{n}\otimes\mathbf{u}^{n+\frac{\gamma}{2}}\right)-\nabla p^{n}\\
&\mathbf{u}^{n+\gamma,*}|_{\partial\Omega}=\mathbf{u}_{D}^{n+\gamma}.
\end{aligned}\tag{4}$$

Notice that, in order to avoid solving a nonlinear system at each time step, an approximation is introduced in the nonlinear momentum advection term, so that $\mathbf{u}^{n+\frac{\gamma}{2}}$ is defined by extrapolation as

$$\mathbf{u}^{n+\frac{\gamma}{2}}=\left(1+\frac{\gamma}{2\left(1-\gamma\right)}\right)\mathbf{u}^{n}-\frac{\gamma}{2\left(1-\gamma\right)}\mathbf{u}^{n-1}.$$

Alternatively, $\mathbf{u}^{n+\frac{\gamma}{2}}$ can be replaced by $\mathbf{u}^{n+\gamma,*}$ in the left hand side and by $\mathbf{u}^{n}$ in the right hand side of (4), respectively, and $\mathbf{u}^{n+\gamma,*}$ can be determined by fixed point iteration. Numerical experiments show that this fully nonlinear formulation is necessary to achieve accurate results for larger Courant number values, see the discussion in Section 4. Following [10], we set then $\delta p^{n+\gamma}=p^{n+\gamma}-p^{n}$ and impose

$$\begin{aligned}
&\frac{\mathbf{u}^{n+\gamma}-\mathbf{u}^{n+\gamma,*}}{\gamma\Delta t}=-\nabla\delta p^{n+\gamma}\\
&\frac{1}{c^{2}}\frac{\delta p^{n+\gamma}}{\gamma\Delta t}+\nabla\cdot\mathbf{u}^{n+\gamma}=0.
\end{aligned}\tag{5}$$

Substituting the first equation into the second in (5), one obtains the Helmholtz equation

$$\frac{1}{c^{2}\gamma^{2}\Delta t^{2}}\delta p^{n+\gamma}-\Delta\delta p^{n+\gamma}=-\frac{1}{\gamma\Delta t}\nabla\cdot\mathbf{u}^{n+\gamma,*},\tag{6}$$

which is solved with the boundary condition $\nabla\delta p^{n+\gamma}\cdot\mathbf{n}|_{\partial\Omega}=0$. Once this equation is solved, the final velocity update for the first stage $\mathbf{u}^{n+\gamma}=\mathbf{u}^{n+\gamma,*}-\gamma\Delta t\nabla\delta p^{n+\gamma}$ can be computed. Notice that the previous procedure is equivalent to introducing the intermediate update $\mathbf{u}^{n+\gamma,**}=\mathbf{u}^{n+\gamma,*}+\gamma\Delta t\nabla p^{n}$, solving

$$\frac{1}{c^{2}}\frac{p^{n+\gamma}}{\gamma^{2}\Delta t^{2}}-\Delta p^{n+\gamma}=-\frac{1}{\gamma\Delta t}\nabla\cdot\mathbf{u}^{n+\gamma,**}+\frac{1}{c^{2}}\frac{p^{n}}{\gamma^{2}\Delta t^{2}}\tag{7}$$

and then setting $\mathbf{u}^{n+\gamma}=\mathbf{u}^{n+\gamma,**}-\gamma\Delta t\nabla p^{n+\gamma}$. The second TR-BDF2 stage is performed in a similar manner. We first define the second momentum predictor:

$$\frac{\mathbf{u}^{n+1,*}-\mathbf{u}^{n+\gamma}}{(1-\gamma)\,\Delta t}-\frac{a_{33}}{Re}\Delta\mathbf{u}^{n+1,*}+a_{33}\,\nabla\cdot\left(\mathbf{u}^{n+1,*}\otimes\mathbf{u}^{n+\frac{3}{2}\gamma}\right)= \tag{8}$$
$$\frac{a_{32}}{Re}\Delta\mathbf{u}^{n+\gamma}-a_{32}\,\nabla\cdot\left(\mathbf{u}^{n+\gamma}\otimes\mathbf{u}^{n+\gamma}\right)+\frac{a_{31}}{Re}\Delta\mathbf{u}^{n}-a_{31}\,\nabla\cdot\left(\mathbf{u}^{n}\otimes\mathbf{u}^{n}\right)-\nabla p^{n+\gamma}$$
$$\mathbf{u}^{n+1,*}|_{\partial\Omega^D}=\mathbf{u}_D^{n+1},$$

where one has

$$a_{31}=\frac{1-\gamma}{2\,(2-\gamma)}\quad a_{32}=\frac{1-\gamma}{2\,(2-\gamma)}\quad a_{33}=\frac{1}{2-\gamma}.$$

Again, in order to avoid solving a nonlinear system at each time step, an approximation is introduced in the nonlinear momentum advection term, so that $\mathbf{u}^{n+\frac{3}{2}\gamma}$ is defined by extrapolation as

$$\mathbf{u}^{n+\frac{3}{2}\gamma}=\left(1+\frac{1+\gamma}{\gamma}\right)\mathbf{u}^{n+\gamma}-\frac{1-\gamma}{\gamma}\mathbf{u}^{n}.$$

Alternatively, $\mathbf{u}^{n+\frac{3}{2}\gamma}$ can be replaced by $\mathbf{u}^{n+1,*}$, which can then be determined by fixed point iteration. We set then $\delta p^{n+1}=p^{n+1}-p^{n+\gamma}$ and impose

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n+1,*}}{(1-\gamma)\Delta t}=-\nabla\delta p^{n+1}$$
$$\frac{1}{c^2}\frac{\delta p^{n+1}}{(1-\gamma)\Delta t}+\nabla\cdot\mathbf{u}^{n+1}=0. \tag{9}$$

Substituting the first equation into the second in (9), one obtains the Helmholtz equation

$$\frac{1}{c^2(1-\gamma)^2\Delta t^2}\delta p^{n+1}-\Delta\delta p^{n+1}=-\frac{1}{(1-\gamma)\Delta t}\,\nabla\cdot\mathbf{u}^{n+1,*}, \tag{10}$$

which is solved with the boundary condition $\nabla\delta p^{n+1}\cdot\mathbf{n}|_{\partial\Omega}=0$. Once this equation is solved, the final velocity update

$$\mathbf{u}^{n+1}=\mathbf{u}^{n+1,*}-(1-\gamma)\Delta t\nabla\delta p^{n+1}$$

can be computed. Also for this second stage, notice that the procedure is equivalent to setting $\mathbf{u}^{n+1,**}=\mathbf{u}^{n+1,*}+(1-\gamma)\Delta t\nabla p^{n+\gamma}$, solving

$$\frac{1}{c^2}\frac{p^{n+1}}{(1-\gamma)^2\Delta t^2}-\Delta p^{n+1}=-\frac{1}{(1-\gamma)\Delta t}\,\nabla\cdot\mathbf{u}^{n+1,**}+\frac{1}{c^2}\frac{p^{n+\gamma}}{(1-\gamma)^2\Delta t^2} \tag{11}$$

and then setting $\mathbf{u}^{n+1}=\mathbf{u}^{n+1,**}-(1-\gamma)\Delta t\nabla p^{n+1}$.

For the purposes of the comparisons that will be reported in Section 4, we also present two alternative and very popular second order projection methods, proposed respectively in [10] and in [33], which are based on the parent methods of TR-BDF2, i.e. the Crank-Nicolson (or Trapezoidal Rule) method and the BDF2 method, respectively. We briefly recall the formulation of these schemes in the framework of the artificial compressibility formulation. The momentum predictor for the Bell-Colella-Glaz [10] projection method reads as follows

$$\begin{aligned}
&\frac{\mathbf{u}^{n+1,*}-\mathbf{u}^n}{\Delta t}-\frac{1}{2Re}\Delta\mathbf{u}^{n+1,*}+\left[(\mathbf{u}\cdot\nabla)\,\mathbf{u}\right]^{n+\frac{1}{2},*}=\\
&\frac{1}{2Re}\Delta\mathbf{u}^n-\nabla p^n\\
&\mathbf{u}^{n+1,*}|_{\partial\Omega}=\mathbf{u}_D^{n+1}.
\end{aligned}\tag{12}$$

Notice that here we have set $\mathbf{u}^{n+\frac{1}{2},*}=\frac{1}{2}\left(\mathbf{u}^{n+1,*}+\mathbf{u}^n\right)$, so that the scheme is fully nonlinear. On the other hand, setting $\delta p^{n+1}=p^{n+1}-p^n$, we obtain the following Helmholtz equation for the projection stage

$$\begin{aligned}
&\frac{1}{c^2\Delta t^2}\delta p^{n+1}-\Delta\delta p^{n+1}=-\frac{1}{\Delta t}\,\nabla\cdot\mathbf{u}^{n+1,*}\\
&\nabla\delta p^{n+1}\cdot\mathbf{n}|_{\partial\Omega}=0.
\end{aligned}\tag{13}$$

Eventually, the velocity has to be updated with the gradient of the pressure increment:

$$\mathbf{u}^{n+1}=\mathbf{u}^{n+1,*}-\Delta t\nabla\delta p^{n+1}.\tag{14}$$

It is apparent that this method is essentially based on the Crank-Nicolson time discretization approach. A method based on the BDF2 scheme has been presented instead by Guermond and Quartapelle in [33]. The momentum predictor reads as follows

$$\begin{aligned}
&\frac{3\mathbf{u}^{n+1,*}-4\mathbf{u}^n+\mathbf{u}^{n-1}}{\Delta t}-\frac{1}{Re}\Delta\mathbf{u}^{n+1,*}+\left(\mathbf{u}^{n,*}\cdot\nabla\right)\mathbf{u}^{n+1,*}\\
&+\frac{1}{2}\left(\nabla\cdot\mathbf{u}^{n,*}\right)\mathbf{u}^{n+1,*}=-\nabla p^n\\
&\mathbf{u}^{n+1,*}|_{\partial\Omega}=\mathbf{u}_D^{n+1}.
\end{aligned}\tag{15}$$

The Helmholtz equation for the projection stage is

$$\begin{aligned}
&\frac{1}{c^2\Delta t^2}p^{n+1}-\Delta p^{n+1}=-\frac{1}{\Delta t}\,\nabla\cdot\mathbf{u}^{n+1,**}+\frac{1}{c^2\Delta t^2}p^n\\
&\nabla\delta p^{n+1}\cdot\mathbf{n}|_{\partial\Omega}=0,
\end{aligned}\tag{16}$$

where $\mathbf{u}^{n+1,**}=\mathbf{u}^{n+1,*}+\frac{2}{3}\Delta t\nabla p^n$. Eventually, the velocity is updated with the gradient of the computed pressure:

$$\mathbf{u}^{n+1}=\mathbf{u}^{n+1,**}-\frac{2}{3}\Delta t\nabla p^{n+1}.\tag{17}$$

3 The spatial discretization

For the spatial discretization, we consider discontinuous finite element approximations, due to their great flexibility in performing mesh adaptation. We consider a decomposition of the domain Ω into a family of hexahedra $\mathcal{T}_h$ (quadrilaterals in the two-dimensional case) and denote each element by K. The skeleton $\mathcal{E}$ denotes the set of all element faces and $\mathcal{E} = \mathcal{E}^I \cup \mathcal{E}^B$, where $\mathcal{E}^I$ is the subset of interior faces and $\mathcal{E}^B$ is the subset of boundary faces. We also introduce the following finite element spaces

$$Q_k = \left\{ v \in L^2(\Omega) : v|_K \in \mathbb{Q}_k \quad \forall K \in \mathcal{T}_h \right\}$$

and

$$\mathbf{Q}_k = [Q_k]^d,$$

where $\mathbb{Q}_k$ is the space of polynomials of degree k in each coordinate direction. Considering the well-posedness analyses in [49, 53], the finite element spaces that will be used for the discretization of velocity and pressure are $\mathbf{V}_h = \mathbf{Q}_k$ and $Q_h = Q_{k-1} \cap L^2_0(\Omega)$, respectively, where $k \geq 2$. Notice that, while for the sake of coherence with the time discretization and of comparison with second order finite volume methods we will mostly consider the case $k = 2$ in the following, the formulation we present is completely general and also the implementation validated in Section 4 supports arbitrary values of k. Suitable jump and average operators can then be defined as customary for finite element discretizations, see e.g. [3]. A face $\Gamma \in \mathcal{E}^I$ shares two elements that we denote by K^+ with outward unit normal $\mathbf{n}^+$ and K^- with outward unit normal $\mathbf{n}^-$, whereas for a face $\Gamma \in \mathcal{E}^B$ we denote by $\mathbf{n}$ the outward unit normal. For a scalar function φ the jump is defined as

$$[[\varphi]] = \varphi^+ \mathbf{n}^+ + \varphi^- \mathbf{n}^- \quad \text{if } \Gamma \in \mathcal{E}^I \qquad [[\varphi]] = \varphi \mathbf{n} \quad \text{if } \Gamma \in \mathcal{E}^B.$$

The average is defined as

$$\{\{\varphi\}\} = \frac{1}{2}\left(\varphi^+ + \varphi^-\right) \quad \text{if } \Gamma \in \mathcal{E}^I \qquad \{\{\varphi\}\} = \varphi \quad \text{if } \Gamma \in \mathcal{E}^B.$$

Similar definitions apply for a vector function $\boldsymbol{\varphi}$:

$$[[\boldsymbol{\varphi}]] = \boldsymbol{\varphi}^+ \cdot \mathbf{n}^+ + \boldsymbol{\varphi}^- \cdot \mathbf{n}^- \quad \text{if } \Gamma \in \mathcal{E}^I \qquad [[\boldsymbol{\varphi}]] = \boldsymbol{\varphi} \cdot \mathbf{n} \quad \text{if } \Gamma \in \mathcal{E}^B$$

$$\{\{\boldsymbol{\varphi}\}\} = \frac{1}{2}\left(\boldsymbol{\varphi}^+ + \boldsymbol{\varphi}^-\right) \quad \text{if } \Gamma \in \mathcal{E}^I \qquad \{\{\boldsymbol{\varphi}\}\} = \boldsymbol{\varphi} \quad \text{if } \Gamma \in \mathcal{E}^B.$$

For vector functions, it is also useful to define a tensor jump as:

$$\langle\langle \boldsymbol{\varphi} \rangle\rangle = \boldsymbol{\varphi}^+ \otimes \mathbf{n}^+ + \boldsymbol{\varphi}^- \otimes \mathbf{n}^- \quad \text{if } \Gamma \in \mathcal{E}^I \qquad \langle\langle \boldsymbol{\varphi} \rangle\rangle = \boldsymbol{\varphi} \otimes \mathbf{n} \quad \text{if } \Gamma \in \mathcal{E}^B.$$

Given these definitions, the weak formulation of the momentum predictor equation for the first stage is obtained multiplying equation (4) by a test function $\mathbf{v} \in \mathbf{V}_h$, integrating over $K \in \mathcal{T}_h$ and applying Green's theorem. To impose the boundary conditions, we set $\left(\mathbf{u}^{n+\gamma,*}\right)^- = -\left(\mathbf{u}^{n+\gamma,*}\right)^+ + 2\mathbf{u}_D^{n+\gamma}$ with $\left[\nabla\left(\mathbf{u}^{n+\gamma,*}\right)^+\right] \cdot \mathbf{n} = \left[\nabla\left(\mathbf{u}^{n+\gamma,*}\right)^-\right] \cdot \mathbf{n}$.

We now treat separately the discretization of the diffusion and advection contributions, respectively. The approximation of the diffusion term is based on the symmetric interior penalty method (SIP) [2]. We denote the scalar product between two second-order tensors by

$$\mathbf{A} : \mathbf{B} = \sum_{i,j} A_{ij} B_{ij}.$$

Following [21], we set for each face Γ of a cell K

$$\sigma^{\mathbf{u}}_{\Gamma,K} = (k+1)^2 \frac{\mathrm{diam}(\Gamma)}{\mathrm{diam}(K)} \tag{18}$$

and we define the penalization constant for the SIP method as

$$C_u = \frac{1}{2}\left(\sigma^{\mathbf{u}}_{\Gamma,K^+} + \sigma^{\mathbf{u}}_{\Gamma,K^-}\right)$$

if $\Gamma \in \mathcal{E}^I$ and $C_u = \sigma^{\mathbf{u}}_{\Gamma,K}$ otherwise. Taking into account boundary conditions as previously discussed and summing over all $K \in \mathcal{T}_h$, we can define the following bilinear form:

$$\begin{aligned}
a^{(1)}_{\mathbf{u}}(\mathbf{u},\mathbf{v}) &= \frac{1}{2Re}\sum_{K\in\mathcal{T}_h}\int_K \nabla\mathbf{u} : \nabla\mathbf{v} d\Omega \\
&- \frac{1}{2Re}\sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma \{\{\nabla\mathbf{u}\}\} : \langle\langle\mathbf{v}\rangle\rangle \, d\Sigma \\
&- \frac{1}{2Re}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma (\nabla\mathbf{u})\,\mathbf{n}\cdot\mathbf{v} d\Sigma \\
&- \frac{1}{2Re}\sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma \langle\langle\mathbf{u}\rangle\rangle : \{\{\nabla\mathbf{v}\}\} \, d\Sigma \\
&- \frac{1}{2Re}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma (\mathbf{u}\otimes\mathbf{n}) : \nabla\mathbf{v} d\Sigma \\
&+ \frac{1}{2Re}\sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma C_u \langle\langle\mathbf{u}\rangle\rangle : \langle\langle\mathbf{v}\rangle\rangle \, d\Sigma \\
&+ \frac{1}{2Re}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma 2C_u (\mathbf{u}\cdot\mathbf{v}) \, d\Sigma.
\end{aligned} \tag{19}$$

The approximation of the advection term employs the widely used local Lax-Friedrichs (LF) flux, see e.g. [29]. Setting

$$\lambda = \max\left(\left|\left(\mathbf{u}^{n+\frac{\gamma}{2}}\right)^+ \cdot \mathbf{n}\right|, \left|\left(\mathbf{u}^{n+\frac{\gamma}{2}}\right)^- \cdot \mathbf{n}\right|\right)$$

with $\mathbf{n} = \mathbf{n}^{\pm}$ and taking into account boundary conditions, we define the trilinear form

$$\begin{aligned} c^{(1)}(\mathbf{u}^{n+\frac{\gamma}{2}}, \mathbf{u}, \mathbf{v}) &= -\frac{1}{2}\sum_{K\in\mathcal{T}_h}\int_K \left(\mathbf{u}\otimes\mathbf{u}^{n+\frac{\gamma}{2}}\right) : \nabla\mathbf{v} d\Omega \\ &+ \frac{1}{2}\sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma \left(\left\{\left\{\mathbf{u}\otimes\mathbf{u}^{n+\frac{\gamma}{2}}\right\}\right\}\right) : \langle\langle\mathbf{v}\rangle\rangle \, d\Sigma \\ &+ \frac{1}{2}\sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma \frac{\lambda}{2}\langle\langle\mathbf{u}\rangle\rangle : \langle\langle\mathbf{v}\rangle\rangle \, d\Sigma \\ &+ \frac{1}{2}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma \lambda\left(\mathbf{u}\cdot\mathbf{v}\right) d\Sigma \end{aligned} \tag{20}$$

Finally, we also define the functional

$$\begin{aligned} F_{\mathbf{u}}^{(1)}(\mathbf{v})^{n+\gamma} = &- \frac{1}{2Re}\sum_{K\in\mathcal{T}_h}\int_K \nabla\mathbf{u}^n : \nabla\mathbf{v} d\Omega + \frac{1}{2Re}\sum_{\Gamma\in\mathcal{E}}\int_\Gamma \{\{\nabla\mathbf{u}^n\}\} : \langle\langle\mathbf{v}\rangle\rangle \, d\Sigma \\ &+ \frac{1}{2}\sum_{K\in\mathcal{T}_h}\int_K \left(\mathbf{u}^n\otimes\mathbf{u}^{n+\frac{\gamma}{2}}\right) : \nabla\mathbf{v} d\Omega \\ &- \frac{1}{2}\sum_{\Gamma\in\mathcal{E}}\int_\Gamma \left(\left\{\left\{\mathbf{u}^n\otimes\mathbf{u}^{n+\frac{\gamma}{2}}\right\}\right\}\right) : \langle\langle\mathbf{v}\rangle\rangle \, d\Sigma \\ &+ \sum_{K\in\mathcal{T}_h}\int_K p^n \nabla\cdot\mathbf{v} d\Omega - \sum_{\Gamma\in\mathcal{E}}\int_\Gamma \{\{p^n\}\}\left[\left[\mathbf{v}\right]\right] d\Sigma \\ &- \frac{1}{2Re}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma \left(\mathbf{u}_D^{n+\gamma}\otimes\mathbf{n}\right) : \nabla\mathbf{v} d\Sigma \\ &+ \frac{1}{2Re}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma 2C_u\left(\mathbf{u}_D^{n+\gamma}\cdot\mathbf{v}\right) d\Sigma \\ &- \frac{1}{2}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma \left(\mathbf{u}_D^{n+\gamma}\otimes\mathbf{u}^{n+\frac{\gamma}{2}}\right)\mathbf{n}\cdot\mathbf{v} d\Sigma \\ &+ \frac{1}{2}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma \lambda\left(\mathbf{u}_D^{n+\gamma}\cdot\mathbf{v}\right) d\Sigma. \end{aligned} \tag{21}$$

which also includes the terms representing the weak form of Dirichlet boundary conditions.

The complete weak formulation of the first stage velocity update reads then as follows: *given* $\mathbf{u}^{n+\frac{\gamma}{2}}, \mathbf{u}^n \in \mathbf{V}_h$ *and* $p^n \in Q_h$, *find* $\mathbf{u}^{n+\gamma,*} \in \mathbf{V}_h$ *such that:*

$$\sum_{K\in\mathcal{T}_h} \int_K \frac{1}{\gamma\Delta t}\mathbf{u}^{n+\gamma,*}\cdot\mathbf{v}d\Omega + a_{\mathbf{u}}^{(1)}(\mathbf{u}^{n+\gamma,*},\mathbf{v}) + c^{(1)}(\mathbf{u}^{n+\frac{\gamma}{2}},\mathbf{u}^{n+\gamma,*},\mathbf{v})$$
$$= \sum_{K\in\mathcal{T}_h} \int_K \frac{1}{\gamma\Delta t}\mathbf{u}^{n}\cdot\mathbf{v}d\Omega + F_{\mathbf{u}}^{(1)}(\mathbf{v})^{n+\gamma} \quad \forall \mathbf{v}\in\mathbf{V}_h. \tag{22}$$

For the projection steps defined by equation (7) we apply again the SIP method. In order to impose homogeneous Neumann boundary conditions we prescribe $\left[\nabla\left(p^{n+\gamma}\right)^-\right]\mathbf{n} = -\left[\nabla\left(p^{n+\gamma}\right)^+\right]\mathbf{n}$: for this reason, no contribution from boundary faces arises. We then multiply by a test function $q \in Q_h$, we apply Green's theorem and we define:

$$\begin{aligned} a_p(p,q) &= \sum_{K\in\mathcal{T}_h}\int_K \nabla p\cdot\nabla q d\Omega - \sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma \left\{\left\{\nabla p^{n+\gamma}\right\}\right\}\cdot[[q]]\,d\Sigma \\ &- \sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma [[p]]\cdot\{\{\nabla q\}\}\,d\Sigma \\ &+ \sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma C_p\,[[p]]\cdot[[q]]\,d\Sigma \qquad (23) \\ F_p^{(1)}(q)^{n+\gamma} &= \sum_{K\in\mathcal{T}_h}\int_K \frac{1}{\gamma\Delta t}\mathbf{u}^{n+\gamma,**}\cdot\nabla q d\Omega \\ &- \sum_{\Gamma\in\mathcal{E}}\int_\Gamma \frac{1}{\gamma\Delta t}\left\{\left\{\mathbf{u}^{n+\gamma,**}\right\}\right\}\cdot[[q]]\,d\Sigma \qquad (24) \end{aligned}$$

and again we set

$$\sigma_{\Gamma,K}^p = k^2\frac{\mathrm{diam}(\Gamma)}{\mathrm{diam}(K)}, \tag{25}$$

while, if $\Gamma \in \mathcal{E}^I$, we set $C_p = \frac{1}{2}\left(\sigma_{\Gamma,K^+}^p + \sigma_{\Gamma,K^-}^p\right)$, otherwise $C_p = \sigma_{\Gamma,K}^p$. The weak formulation of equation (7) reads then: *given* $p^n \in Q_h$, *find* $p^{n+\gamma} \in Q_h$ *such that*

$$\sum_{K\in\mathcal{T}_h}\int_K \frac{1}{c^2\gamma^2\Delta t^2}p^{n+\gamma}q d\Omega + a_p(p^{n+\gamma},q)$$
$$= \sum_{K\in\mathcal{T}_h}\int_K \frac{1}{c^2\gamma^2\Delta t^2}p^{n}q d\Omega + F_p^{(1)}(q)^{n+\gamma} \qquad \forall q\in Q_h. \tag{26}$$

The second stage can be described in a similar manner. We start defining

the bilinear forms for the second momentum predictor as

$$
\begin{aligned}
a_{\mathbf{u}}^{(2)}(\mathbf{u},\mathbf{v}) = & \frac{a_{33}}{Re}\sum_{K\in\mathcal{T}_h}\int_K \nabla\mathbf{u} : \nabla\mathbf{v} d\Omega \\
& - \frac{a_{33}}{Re}\sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma \{\{\nabla\mathbf{u}\}\} : \langle\langle\mathbf{v}\rangle\rangle \, d\Sigma \\
& - \frac{a_{33}}{Re}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma (\nabla\mathbf{u})\,\mathbf{n}\cdot\mathbf{v} d\Sigma \\
& - \frac{a_{33}}{Re}\sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma \langle\langle\mathbf{u}\rangle\rangle : \{\{\nabla\mathbf{v}\}\} \, d\Sigma \\
& - \frac{a_{33}}{Re}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma (\mathbf{u}\otimes\mathbf{n}) : \nabla\mathbf{v} d\Sigma \\
& + \frac{a_{33}}{Re}\sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma C_u \langle\langle\mathbf{u}\rangle\rangle : \langle\langle\mathbf{v}\rangle\rangle \, d\Sigma \\
& + \frac{a_{33}}{Re}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma 2C_u (\mathbf{u}\cdot\mathbf{v}) \, d\Sigma
\end{aligned} \tag{27}
$$

$$
\begin{aligned}
c^{(2)}(\mathbf{u}^{n+\frac{3}{2}\gamma},\mathbf{u},\mathbf{v}) = & - a_{33}\sum_{K\in\mathcal{T}_h}\int_K \left(\mathbf{u}\otimes\mathbf{u}^{n+\frac{3}{2}\gamma}\right) : \nabla\mathbf{v} d\Omega \\
& + a_{33}\sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma \left(\left\{\left\{\mathbf{u}\otimes\mathbf{u}^{n+\frac{3}{2}\gamma}\right\}\right\}\right) : \langle\langle\mathbf{v}\rangle\rangle \, d\Sigma \\
& + a_{33}\sum_{\Gamma\in\mathcal{E}^I}\int_\Gamma \frac{\lambda}{2}\langle\langle\mathbf{u}\rangle\rangle : \langle\langle\mathbf{v}\rangle\rangle \, d\Sigma \\
& + a_{33}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma \lambda (\mathbf{u}\cdot\mathbf{v}) \, d\Sigma,
\end{aligned} \tag{28}
$$

where $\lambda = \max\left(\left|\left(\mathbf{u}^{n+\frac{3}{2}\gamma}\right)^+\cdot\mathbf{n}\right|, \left|\left(\mathbf{u}^{n+\frac{3}{2}\gamma}\right)^-\cdot\mathbf{n}\right|\right)$ with $\mathbf{n}=\mathbf{n}^{\pm}$. We also

define the linear functional:

$$
\begin{aligned}
F_{\mathbf{u}}^{(2)}(\mathbf{v})^{n+1} = &- \frac{a_{32}}{Re}\sum_{K\in\mathcal{T}_h}\int_K \nabla\mathbf{u}^{n+\gamma} : \nabla\mathbf{v}d\Omega + \frac{a_{32}}{Re}\sum_{\Gamma\in\mathcal{E}}\int_\Gamma \left\{\left\{\nabla\mathbf{u}^{n+\gamma}\right\}\right\} : \langle\langle\mathbf{v}\rangle\rangle\, d\Sigma \\
&- \frac{a_{31}}{Re}\sum_{K\in\mathcal{T}_h}\int_K \nabla\mathbf{u}^{n} : \nabla\mathbf{v}d\Omega + \frac{a_{31}}{Re}\sum_{\Gamma\in\mathcal{E}}\int_\Gamma \{\{\nabla\mathbf{u}^{n}\}\} : \langle\langle\mathbf{v}\rangle\rangle\, d\Sigma \\
&+ a_{32}\sum_{K\in\mathcal{T}_h}\int_K \left(\mathbf{u}^{n+\gamma}\otimes\mathbf{u}^{n+\gamma}\right) : \nabla\mathbf{v}d\Omega \\
&- a_{32}\sum_{\Gamma\in\mathcal{E}}\int_\Gamma \left(\left\{\left\{\mathbf{u}^{n+\gamma}\otimes\mathbf{u}^{n+\gamma}\right\}\right\}\right) : \langle\langle\mathbf{v}\rangle\rangle\, d\Sigma \\
&+ a_{31}\sum_{K\in\mathcal{T}_h}\int_K \left(\mathbf{u}^{n}\otimes\mathbf{u}^{n}\right) : \nabla\mathbf{v}d\Omega - a_{31}\sum_{\Gamma\in\mathcal{E}}\int_\Gamma \left(\{\{\mathbf{u}^{n}\otimes\mathbf{u}^{n}\}\}\right) : \langle\langle\mathbf{v}\rangle\rangle\, d\Sigma \\
&+ \sum_{K\in\mathcal{T}_h}\int_K p^{n+\gamma}\nabla\cdot\mathbf{v}d\Omega - \sum_{\Gamma\in\mathcal{E}}\int_\Gamma \left\{\left\{p^{n+\gamma}\right\}\right\}[[\mathbf{v}]]\, d\Sigma \\
&- \frac{a_{33}}{Re}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma \left(\mathbf{u}_D^{n+1}\otimes\mathbf{n}\right) : \nabla\mathbf{v}d\Sigma \\
&+ \frac{a_{33}}{Re}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma 2C_u\left(\mathbf{u}_D^{n+1}\cdot\mathbf{v}\right)d\Sigma \\
&- a_{33}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma \left(\mathbf{u}_D^{n+1}\otimes\mathbf{u}^{n+\frac{3}{2}\gamma}\right)\mathbf{n}\cdot\mathbf{v}d\Sigma \\
&+ a_{33}\sum_{\Gamma\in\mathcal{E}^B}\int_\Gamma \lambda\left(\mathbf{u}_D^{n+1}\cdot\mathbf{v}\right)d\Sigma. \qquad (29)
\end{aligned}
$$

Finally, the weak formulation for the equation (8) reads as follows: *given* $\mathbf{u}^{n+\frac{3}{2}\gamma}, \mathbf{u}^{n+\gamma}\in\mathbf{V}_h$ *and* $p^{n+\gamma}\in Q_h$, *find* $\mathbf{u}^{n+\gamma,*}\in\mathbf{V}_h$ *such that:*

$$
\begin{aligned}
&\sum_{K\in\mathcal{T}_h}\int_K \frac{1}{(1-\gamma)\,\Delta t}\mathbf{u}^{n+1,*}\cdot\mathbf{v}d\Omega + a_{\mathbf{u}}^{(2)}(\mathbf{u}^{n+1,*},\mathbf{v}) + c^{(2)}(\mathbf{u}^{n+\frac{3}{2}\gamma},\mathbf{u}^{n+1,*},\mathbf{v}) \\
&= \sum_{K\in\mathcal{T}_h}\int_K \frac{1}{(1-\gamma)\,\Delta t}\mathbf{u}^{n+\gamma}\cdot\mathbf{v}d\Omega + F_{\mathbf{u}}^{(2)}(\mathbf{v})^{n+1} \quad \forall\mathbf{v}\in\mathbf{V}_h. \qquad (30)
\end{aligned}
$$

We can then immediately define the functional associated to the second projection step as

$$
\begin{aligned}
F_p^{(2)}(q)^{n+1} = &\sum_{K\in\mathcal{T}_h}\int_K \frac{1}{(1-\gamma)\,\Delta t}\mathbf{u}^{n+1,**}\cdot\nabla q d\Omega \\
&- \sum_{\Gamma\in\mathcal{E}}\int_\Gamma \frac{1}{(1-\gamma)\,\Delta t}\left\{\left\{\mathbf{u}^{n+1,**}\right\}\right\}\cdot[[q]]\, d\Sigma. \qquad (31)
\end{aligned}
$$

Therefore, the weak formulation for (11) reads as follows: *given* $p^n \in Q_h$, *find* $p^{n+1} \in Q_h$ *such that:*

$$\sum_{K\in\mathcal{T}_h} \int_K \frac{1}{c^2(1-\gamma)^2\Delta t^2} p^{n+1} q d\Omega + a(p^{n+1}, q)^{n+1} =$$
$$\sum_{K\in\mathcal{T}_h} \int_K \frac{1}{c^2(1-\gamma)^2\Delta t^2} p^{n+\gamma} q d\Omega + F_p^{(2)}(q)^{n+1} \qquad \forall q \in Q_h. \tag{32}$$

We now derive the fully discrete algebraic expressions corresponding to each of the two stages. We denote by $\boldsymbol{\varphi}_i(\mathbf{x})$ the basis functions for the space $\mathbf{V}_h$ and by $\psi_i(\mathbf{x})$ the basis functions for the space Q_h, respectively, so that the discrete approximations of $\mathbf{u}$ and p read as follows

$$\mathbf{u} \approx \mathbf{u}_h = \sum_{j=1}^{\dim(\mathbf{V}_h)} u_j(t)\boldsymbol{\varphi}_j(\mathbf{x}) \qquad p \approx p_h = \sum_{j=1}^{\dim(Q_h)} p_j(t)\psi_j(\mathbf{x}).$$

For the first stage, we take $\mathbf{v} = \boldsymbol{\varphi}_i$, $i = 1, \dots, \dim(\mathbf{V}_h)$ and we exploit the representation introduced above to obtain the matrices

$$M_{ij} = \sum_{K\in\mathcal{T}_h} \int_K \boldsymbol{\varphi}_j \cdot \boldsymbol{\varphi}_i d\Omega \tag{33}$$
$$A_{ij}^{n+\gamma} = a_{\mathbf{u}}^{(1)}\left(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_j\right) \tag{34}$$
$$C_{ij}\left(\mathbf{u}^{n+\frac{\gamma}{2}}\right) = c^{(1)}\left(\mathbf{u}^{n+\frac{\gamma}{2}}, \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i\right) \tag{35}$$

After computing the integrals in the previous formulae by appropriate quadrature rules, one obtains the algebraic system

$$\left(\frac{1}{\gamma\Delta t}\mathbf{M} + \mathbf{A}^{n+\gamma} + \mathbf{C}\left(\mathbf{u}^{n+\frac{\gamma}{2}}\right)\right)\mathbf{U}_h^{n+\gamma,*} = \frac{1}{\gamma\Delta t}\mathbf{M}\mathbf{U}_h^n + \mathbf{F}_{\mathbf{u}}^{n+\gamma}, \tag{36}$$

where $\mathbf{U}_h$ denotes the vector of the discrete degrees of freedom associated to the velocity field and $\mathbf{F}_{\mathbf{u}}^{n+\gamma}$ is the vector obtained evaluating $F_{\mathbf{u}}^{(1)}(\boldsymbol{\varphi}_i)^{n+\gamma}, i = 1, \dots, \dim(\mathbf{V}_h)$. The same procedure can be applied for the projection step, obtaining the matrices

$$M_{ij}^p = \sum_{K\in\mathcal{T}_h} \int_K \psi_j \psi_i d\Omega \tag{37}$$

$$K_{ij} = a_p(\psi_j, \psi_i). \tag{38}$$

After computing the integrals in the previous formulae by appropriate quadrature rules, one obtains the algebraic counterpart of (26)

$$\left(\frac{1}{c^2\gamma^2\Delta t^2}\mathbf{M}^p + \mathbf{K}\right)\mathbf{P}_h^{n+\gamma} = \frac{1}{c^2\gamma^2\Delta t^2}\mathbf{M}^p\mathbf{P}_h^n + \mathbf{F}_p^{n+\gamma} \tag{39}$$

where again $\mathbf{P}_h$ denotes the vector of the discrete degrees of freedom associated to pressure and $\mathbf{F}_p^{n+\gamma}$ is the vector obtained evaluating $F_p^{(1)}(\boldsymbol{\psi}_i)^{n+\gamma}, i = 1, \ldots, \dim(Q_h)$. For the second stage, we proceed in a similar manner; for the momentum predictor (30) we obtain

$$\begin{aligned}\left(\frac{1}{(1-\gamma)\,\Delta t}\mathbf{M} + \mathbf{A}^{n+1} + \mathbf{C}\left(\mathbf{u}^{n+\frac{3}{2}\gamma}\right)\right)\mathbf{U}_h^{n+1,*} &= \frac{1}{(1-\gamma)\,\Delta t}\mathbf{M}\mathbf{U}_h^{n+\gamma} \\ &+ \mathbf{F}_{\mathbf{u}}^{n+1},\end{aligned} \tag{40}$$

where we set

$$A_{ij}^{n+1} = a_{\mathbf{u}}^{(2)}\left(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i\right) \tag{41}$$

$$C_{ij}\left(\mathbf{u}^{n+\frac{3}{2}\gamma}\right) = c^{(2)}\left(\mathbf{u}^{n+\frac{3}{2}\gamma}, \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i\right) \tag{42}$$

and $\mathbf{F}_{\mathbf{u}}^{n+1}$ is the vector obtained evaluating $F_{\mathbf{u}}^{(2)}(\boldsymbol{\varphi}_i)^{n+1}, i = 1...\dim(\mathbf{V}_h)$. Eventually, as algebraic counterpart of (32) we obtain

$$\left(\frac{1}{c^2\,(1-\gamma)^2\,\Delta t^2}\mathbf{M}^p + \mathbf{K}\right)\mathbf{P}_h^{n+1} = \frac{1}{c^2\,(1-\gamma)^2\,\Delta t^2}\mathbf{M}^p\mathbf{P}_h^{n+\gamma} + \mathbf{F}_p^{n+1} \tag{43}$$

where again $\mathbf{F}_p^{n+1}$ is the vector obtained evaluating $F_p^{(2)}(\boldsymbol{\psi}_i)^{n+1}, i = 1...\dim(Q_h)$. Notice that, in the evaluation of $\mathbf{F}_p^{n+\gamma}$ and $\mathbf{F}_p^{n+1}$, there is also a preliminary stage which is the projection of ∇p^n and $\nabla p^{n+\gamma}$ into $\mathbf{V}_h$ to compute $\mathbf{u}^{n+\gamma,**}$ and $\mathbf{u}^{n+1,**}$, respectively. In particular, we define the projection matrix $\mathbf{P}$

$$P_{ij} = \sum_{K\in\mathcal{T}_h}\int_K \nabla\psi_j \cdot \boldsymbol{\varphi}_i d\Omega \tag{44}$$

and we solve the linear systems $\mathbf{M}\tilde{\mathbf{u}}^{n+\gamma,**} = \mathbf{P}\mathbf{p}^n$ for the first stage and $\mathbf{M}\tilde{\mathbf{u}}^{n+1,**} = \mathbf{P}\mathbf{p}^{n+\gamma}$, where $\tilde{\mathbf{u}}^{n+\gamma,**}$ and $\tilde{\mathbf{u}}^{n+1,**}$ denote the two required projections. The same procedure has to applied also in the final update of the velocity; in particular, for the first stage we set

$$\mathbf{u}^{n+\gamma} = \mathbf{u}^{n+\gamma,*} - \gamma\Delta t\left(\tilde{\mathbf{u}}^{n+1,**} - \tilde{\mathbf{u}}^{n+\gamma,**}\right) \tag{45}$$

while for the second stage we solve $\mathbf{M}\tilde{\mathbf{u}}^{n+1} = \mathbf{P}\mathbf{p}^{n+1}$ and then we compute

$$\mathbf{u}^{n+1} = \mathbf{u}^{n+1,*} - (1-\gamma)\,\Delta t\left(\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^{n+1,**}\right). \tag{46}$$

4 Numerical experiments

The numerical method outlined in the previous Sections has been validated in a number of relevant benchmarks. Notice that, following e.g. [54, 55], we set $\mathcal{H} = \min\{\text{diam}(K) | K \in \mathcal{T}_h\}$ and we define the stability parameters:

$$C = kU\Delta t/\mathcal{H}, \qquad \mu = k^2\Delta t/(Re\mathcal{H}^2) \tag{47}$$

where U is the magnitude of a characteristic velocity and μ defines the typical stability parameter in the discretization of parabolic terms. We also recall here that k is the polynomila degree of the finite element space chosen for the discretization of the velocity. As stated before, the proposed method has been implemented using the numerical library *deal.II*, which is based on a matrix-free approach [5]. As a consequence, no global sparse matrix is built and only the action of the linear operators defined in Section 3 on a vector is actually implemented. Another feature of the library employed during the numerical simulations is the mesh adaptation capability, as we will see in the presentation of the results. In the following tests, unless differently stated, we take $c = 10^3\ m/s$, which is the order of magnitude of the speed of sound in water. Moreover, the preconditioned conjugate gradient method implemented in the function *SolverCG* of the *deal II* library was employed to solve the Helmholtz equations, while the GMRES solver for the momentum equations is implemented in the function *SolverGMRES* of the same library. A Jacobi preconditioner is used for the two momentum predictors, whereas a Geometric Multigrid preconditioner is employed for the Helmholtz equations.

4.1 Case tests with analytical solution

In order to verify the correctness of our implementation and to assess the convergence property of the scheme, we first perform numerical convergence studies in two and three dimensions, respectively. In two dimensions, we consider as a benchmark the classical Taylor-Green vortex [31] in the box $\Omega = (0, 2\pi)^2$, for which an analytical solution is available:

$$\mathbf{u}(\mathbf{x}, t) = \begin{pmatrix} \cos(x_1)\sin(x_2)e^{-\frac{2t}{Re}} \\ -\sin(x_1)\cos(x_2)e^{-\frac{2t}{Re}} \end{pmatrix} \tag{48}$$

$$p(\mathbf{x}, t) = -\frac{1}{4}\left(\cos(2x_1) + \cos(2x_2)\right)e^{-\frac{4t}{Re}}. \tag{49}$$

In three dimensions, an analogous study has been carried out for the Arnoldi-Beltrami-Childress (ABC) flow, see e.g. [25], whose exact solution is

$$\mathbf{u}(\mathbf{x},t) = \begin{pmatrix} (\sin(x_3)+\cos(x_2))\, e^{-\frac{t}{Re}} \\ (\sin(x_1)+\cos(x_3))\, e^{-\frac{t}{Re}} \\ (\sin(x_2)+\cos(x_1))\, e^{-\frac{t}{Re}} \end{pmatrix}. \tag{50}$$

$$p(\mathbf{x},t) = -\sin(x_1)\cos(x_3) - \sin(x_2)\cos(x_1) - \sin(x_3)\cos(x_2). \tag{51}$$

For the two dimensional case, we performed a convergence test at $T = 3.2$ for $Re = 100$ starting with an initial Cartesian mesh of 8×8 elements and doubling several times the number of elements N_{el} in each direction. The time step was chosen so as to keep $C = 1.63$ constant for all resolutions (hyperbolic scaling), so as to test the accuracy of the method for values of the time steps beyond the stability limit of explicit schemes but not large enough to affect the second order accuracy. The results for the $\mathbf{Q}_2 - Q_1$ and $\mathbf{Q}_3 - Q_2$ cases are reported in Tables 1, 2 and 3, 4, respectively. It can be observed that the expected convergence rates are recovered, without the necessity of employing fixed point iterations to determine the velocity in the two stages. Analogous results are obtained, see Table 5, 6 if distorted meshes with analogous characteristics are employed.

Δt	N_{el}	μ	H^1 rel. error $\mathbf{u}$	H^1 rate $\mathbf{u}$	L^2 rel. error $\mathbf{u}$	L^2 rate $\mathbf{u}$
0.64	8	0.04	1.5		0.38	
0.32	16	0.08	0.65	1.22	0.095	2.01
0.16	32	0.17	0.12	2.45	0.016	2.58
0.08	64	0.33	0.023	2.38	0.0031	2.37

Table 1: Convergence test for the Green-Taylor vortex benchmark computed at $C = 1.63$ with $\mathbf{Q}_2 - Q_1$ elements, relative errors for the velocity in H^1 and L^2 norms.

Δt	N_{el}	μ	L^2 rel. error p	L^2 rate p
0.64	8	0.04	0.43	
0.32	16	0.08	0.14	1.60
0.16	32	0.17	0.04	1.72
0.08	64	0.33	0.011	1.91

Table 2: Convergence test for the Green-Taylor vortex benchmark computed at $C = 1.63$ with $\mathbf{Q}_2 - Q_1$ elements, relative errors for the pressure in L^2 norm.

Δt	N_{el}	μ	H^1 rel. error $\mathbf{u}$	H^1 rate $\mathbf{u}$	L^2 rel. error $\mathbf{u}$	L^2 rate $\mathbf{u}$
0.43	8	0.06	0.28		0.062	
0.21	16	0.12	0.033	3.12	0.0068	3.18
0.11	32	0.25	0.0044	2.88	0.00044	3.93
0.053	64	0.50	0.00059	2.92	0.000031	3.85

Table 3: Convergence test for the Green-Taylor vortex benchmark computed at $C = 1.63$ with $\mathbf{Q}_3 - Q_2$ elements, relative errors for the velocity in H^1 and L^2 norms.

Δt	N_{el}	μ	L^2 rel. error p	L^2 rate p
0.43	8	0.06	0.087	
0.21	16	0.12	0.011	2.93
0.11	32	0.25	0.00075	3.92
0.053	64	0.50	0.000029	4.72

Table 4: Convergence test for the Green-Taylor vortex benchmark computed at $C = 1.63$ with $\mathbf{Q}_3 - Q_2$ elements, relative errors for the pressure in L^2 norm.

Δt	N_{el}	μ	H^1 rel. error $\mathbf{u}$	H^1 rate $\mathbf{u}$	L^2 rel. error $\mathbf{u}$	L^2 rate $\mathbf{u}$
0.52	8	0.05	1.2		0.32	
0.22	16	0.12	0.55	1.13	0.081	1.96
0.11	32	0.25	0.12	2.16	0.012	2.77
0.052	64	0.50	0.021	2.51	0.0023	2.41

Table 5: Convergence test for the Green-Taylor vortex benchmark computed on a distorted mesh at $C = 1.63$ with $\mathbf{Q}_2 - Q_1$ elements, relative errors for the velocity in H^1 and L^2 norms.

Δt	N_{el}	μ	L^2 rel. error p	L^2 rate p
0.52	8	0.05	0.43	
0.22	16	0.12	0.077	2.02
0.11	32	0.25	0.024	1.68
0.052	64	0.50	0.0064	1.91

Table 6: Convergence test for the Green-Taylor vortex benchmark computed on a distorted mesh at $C = 1.63$ with $\mathbf{Q}_2 - Q_1$ elements, relative errors for the pressure in L^2 norm.

The same test was repeated, for the case of $\mathbf{Q}_2 - Q_1$ elements, using the alternative methods [10, 33] summarized in Section 2. It can be observed that, while the convergence rates are analogous, the relative errors in the L^2 norm are about 50% smaller for the TR-BDF2 solver.

Δt	N_{el}	μ	H^1 rel. error $\mathbf{u}$	H^1 rate $\mathbf{u}$	L^2 rel. error $\mathbf{u}$	L^2 rate $\mathbf{u}$
0.64	8	0.04	1.49		0.48	
0.32	16	0.08	0.73	1.04	0.13	1.84
0.16	32	0.17	0.14	2.34	0.0311	2.11
0.08	64	0.33	0.03	2.25	0.0070	2.16

Table 7: Convergence test for the Green-Taylor vortex benchmark computed at $C = 1.63$ with $\mathbf{Q}_2 - Q_1$ elements and the projection method of [10], relative errors for the velocity in H^1 and L^2 norms.

Δt	N_{el}	μ	L^2 rel. error p	L^2 rate p
0.64	8	0.04	0.41	
0.32	16	0.08	0.12	1.75
0.16	32	0.17	0.037	1.75
0.08	64	0.33	0.0087	2.08

Table 8: Convergence test for the Green-Taylor vortex benchmark computed at $C = 1.63$ with $\mathbf{Q}_2 - Q_1$ elements and the projection method of [10], relative errors for the pressure in L^2 norm.

Δt	N_{el}	μ	H^1 rel. error $\mathbf{u}$	H^1 rate $\mathbf{u}$	L^2 rel. error $\mathbf{u}$	L^2 rate $\mathbf{u}$
0.64	8	0.04	0.89		0.28	
0.32	16	0.08	0.40	1.15	0.09	1.69
0.16	32	0.17	0.085	2.24	0.023	1.92
0.08	64	0.33	0.029	1.57	0.0077	1.60

Table 9: Convergence test for the Green-Taylor vortex benchmark computed at $C = 1.63$ with $\mathbf{Q}_2 - Q_1$ elements and the projection method of [33], relative errors for the velocity in H^1 and L^2 norms.

Δt	N_{el}	μ	L^2 rel. error p	L^2 rate p
0.64	8	0.04	0.41	
0.32	16	0.08	0.10	2.06
0.16	32	0.17	0.026	1.91
0.08	64	0.33	0.0067	1.97

Table 10: Convergence test for the Green-Taylor vortex benchmark computed at $C = 1.63$ with $\mathbf{Q}_2 - Q_1$ elements and the projection method of [33], relative errors for the pressure in L^2 norm.

As mentioned in Section 2, when we increase the Courant number, also the TR-BDF2 scheme requires fixed point iterations in the momentum predictor stages in order to preserve its accuracy. As it can be noticed in Tables 11, 12 that the second order convergence rate is still maintained.

Δt	N_{el}	μ	H^1 rel. error $\mathbf{u}$	H^1 rate $\mathbf{u}$	L^2 rel. error $\mathbf{u}$	L^2 rate $\mathbf{u}$
1.18	8	0.08	1.33		0.39	
0.59	16	0.15	0.63	1.07	0.11	1.79
0.29	32	0.31	0.12	2.35	0.028	2.02
0.15	64	0.61	0.028	2.17	0.0059	2.23

Table 11: Convergence test for the Green-Taylor vortex benchmark computed at $C = 3$ with $\mathbf{Q}_2 - Q_1$ elements, relative errors for the velocity in H^1 and L^2 norms.

Δt	N_{el}	μ	L^2 rel. error p	L^2 rate p
1.18	8	0.08	0.49	
0.59	16	0.15	0.13	1.87
0.29	32	0.31	0.04	1.60
0.15	64	0.61	0.013	1.75

Table 12: Convergence test for the Green-Taylor vortex benchmark computed at $C = 3$ with $\mathbf{Q}_2 - Q_1$ elements, relative errors for the pressure in L^2 norm.

For the three dimensional case, an analogous convergence test was performed again at $T = 3.2$ but using $Re = 1$, due to the stability characteristics of the ABC flow, see e.g. the discussion in [25]. We have considered an initial Cartesian mesh of $8 \times 8 \times 8$ elements and we have refined the mesh by doubling each time the number of elements N_{el} in each direction, while keeping $C = 1.63$ constant (hyperbolic scaling). The results for the $\mathbf{Q}_2 - Q_1$ and $\mathbf{Q}_3 - Q_2$ cases are reported in Tables 13, 14 and 15, 16 respectively. It can be observed that the expected convergence rates are recovered for the lower

degree case, also in this case without the necessity of fixed point iterations, while less accurate results are obtained in the higher degree case. Since in this case the problem is diffusion dominated, rather than advection dominated, the loss of accuracy can be readily explained by the very large values obtained in this test for the parabolic stability parameter μ. Repeating the test at constant μ (parabolic scaling), one obtains the results displayed in Tables 17, 18 and 19, 20, which show a clear improvement both in errors and convergence rates.

Δt	N_{el}	μ	H^1 rel. error $\mathbf{u}$	H^1 rate $\mathbf{u}$	L^2 rel. error $\mathbf{u}$	L^2 rate $\mathbf{u}$
0.32	8	2.08	0.019		0.0078	
0.16	16	4.15	0.0054	1.85	0.0022	1.86
0.08	32	8.30	0.0014	1.98	0.00056	1.99
0.04	64	16.60	0.00036	1.91	0.00017	1.75

Table 13: Convergence test for the ABC flow benchmark computed at $C = 1.63$ with $\mathbf{Q}_2 - Q_1$ elements, relative errors for the velocity in H^1 and L^2 norms.

Δt	N_{el}	μ	L^2 rel. error p	L^2 rate p
0.32	8	2.08	1.0	
0.16	16	4.15	0.13	2.93
0.08	32	8.30	0.039	1.74
0.04	64	16.60	0.011	1.79

Table 14: Convergence test for the ABC flow benchmark computed at $C = 1.63$ with $\mathbf{Q}_2 - Q_1$ elements, relative errors for the pressure in L^2 norm.

Δt	N_{el}	μ	H^1 rel. error $\mathbf{u}$	H^1 rate $\mathbf{u}$	L^2 rel. error $\mathbf{u}$	L^2 rate $\mathbf{u}$
0.21	8	3.11	0.0036		0.0019	
0.11	16	6.23	0.0010	1.80	0.0068	2.05
0.053	32	12.45	0.00037	1.5	0.00014	1.68

Table 15: Convergence test for the ABC flow benchmark computed at $C = 1.63$ with $\mathbf{Q}_3 - Q_2$ elements, relative errors for the velocity in H^1 and L^2 norms.

Δt	N_{el}	μ	L^2 rel. error p	L^2 rate p
0.21	8	3.11	0.25	
0.11	16	6.23	0.033	2.93
0.053	32	12.45	0.0097	1.72

Table 16: Convergence test for the ABC flow benchmark computed at $C = 1.63$ with $\mathbf{Q}_3 - Q_2$ elements, relative errors for the pressure in L^2 norm.

Δt	N_{el}	C	H^1 rel. error $\mathbf{u}$	H^1 rate $\mathbf{u}$	L^2 rel. error $\mathbf{u}$	L^2 rate $\mathbf{u}$
0.32	8	1.57	0.019		0.0071	
0.08	16	0.79	0.0045	2.05	0.0013	2.5
0.02	32	0.39	0.0012	1.97	0.00031	2.02
0.005	64	0.20	0.00029	1.98	0.000053	2.54

Table 17: Convergence test for the ABC flow benchmark computed at $\mu = 2$ with $\mathbf{Q}_2 - Q_1$ elements, relative errors for the velocity in H^1 and L^2 norms.

Δt	N_{el}	C	L^2 rel. error p	L^2 rate p
0.32	8	1.57	1.0	
0.08	16	0.79	0.16	2.66
0.02	32	0.39	0.042	1.93
0.005	64	0.20	0.011	1.94

Table 18: Convergence test for the ABC flow benchmark computed at $\mu = 2$ with $\mathbf{Q}_2 - Q_1$ elements, relative errors for the pressure in L^2 norm.

Δt	N_{el}	C	H^1 rel. error $\mathbf{u}$	H^1 rate $\mathbf{u}$	L^2 rel. error $\mathbf{u}$	L^2 rate $\mathbf{u}$
0.14	8	1.05	0.0025		0.00089	
0.034	16	0.52	0.00024	2.70	0.00011	3.08
0.0086	32	0.26	0.000071	1.78	0.000018	2.51

Table 19: Convergence test for the ABC flow benchmark computed at $\mu = 2$ with $\mathbf{Q}_3 - Q_2$ elements, relative errors for the velocity in H^1 and L^2 norms.

Δt	N_{el}	C	L^2 rel. error p	L^2 rate p
0.14	8	1.05	0.20	
0.034	16	0.52	0.027	2.89
0.0086	32	0.26	0.0043	2.65

Table 20: Convergence test for the ABC flow benchmark computed at $\mu = 2$ with $\mathbf{Q}_3 - Q_2$ elements, relative errors for the pressure in L^2 norm.

4.2 Two-dimensional lid driven cavity

The lid driven cavity flow is a classical benchmark for the two-dimensional incompressible Navier-Stokes equations. Reference solutions obtained with high order techniques are reported, among many others, in [4, 15, 16]. For this two-dimensional problem, is it customary to represent the flow also in terms of the streamfunction Ψ, which is defined as the solution of the Laplace problem

$$\begin{aligned} -\Delta\Psi &= \nabla \times \mathbf{u} = \omega \qquad \text{in } \Omega \\ \Psi|_{\partial\Omega} &= 0 \end{aligned} \tag{52}$$

where the symbol $\nabla\times$ denotes the curl operator and the vorticity is the scalar field defined as

$$\omega = \frac{\partial v}{\partial x_1} - \frac{\partial u}{\partial x_2}.$$

We consider the case $Re = 1000$ computed with $\mathbf{Q}_2 - Q_1$ elements on a Cartesian mesh composed of $N_e = 128$ square elements in each coordinate direction, with a time step chosen so that the Courant number is approximately 1.3. The computation is performed until the steady state is reached up to a tolerance of 10^{-7}, which occurs around $T = 70$. The streamfunction contours at steady state are shown in Figure 1 using the same isoline values as in [16]. It can be observed that all the main flow structures are correctly reproduced.

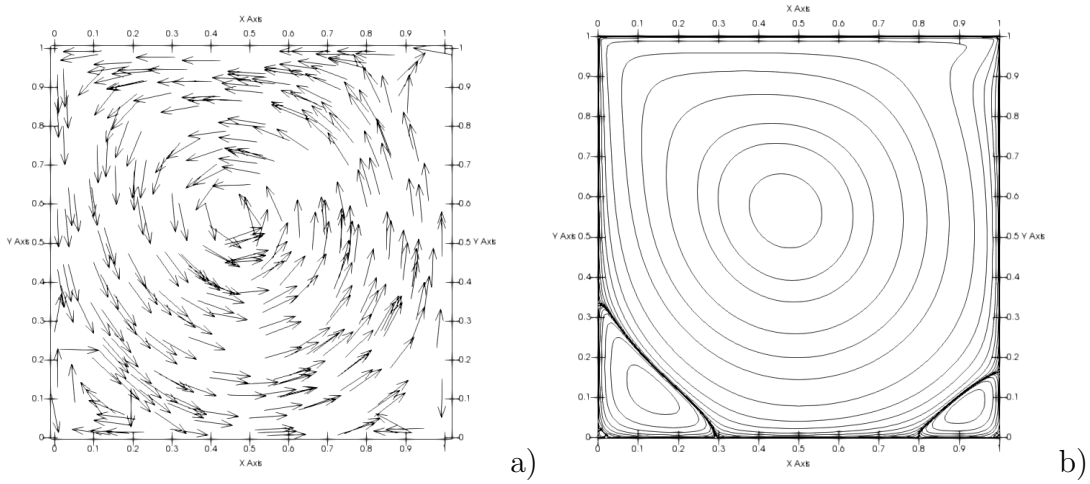


Figure 1: Lid driven cavity benchmark at $Re = 1000$: a) flow field, b) streamfunction contours. Contour values are chosen as in [16].

For a more quantitative comparison, we report in Figure 2 the u component of the velocity and the vorticity ω along the middle of the cavity, together with the reference results of [15]. Good agreement with the reference solution is achieved. The maximum horizontal velocity along the centerline was computed as $u_{max} = 0.3732$ which implies a relative error with respect to the reference solution of the order of 10^{-2}. The vorticity value at the center of the cavity was computed as $\omega_{cen} = 1.9594$, which implies again a relative error with respect to the reference solution of the order of 10^{-2}. For comparison, the same test was repeated also using for the time discretization the parent methods described in [10], [33]. The results are plotted in Figure 3, highlighting the better performance of the proposed method based on TR-BDF2.

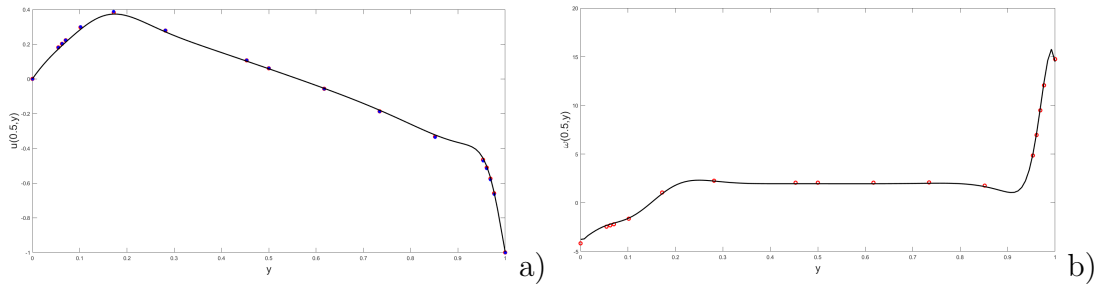


Figure 2: Lid driven cavity benchmark at $Re = 1000$: a) u velocity component values along the middle of the cavity, b) ω values along the middle of the cavity. The continuous line denotes the numerical solution and the dots the reference solution values from [15].

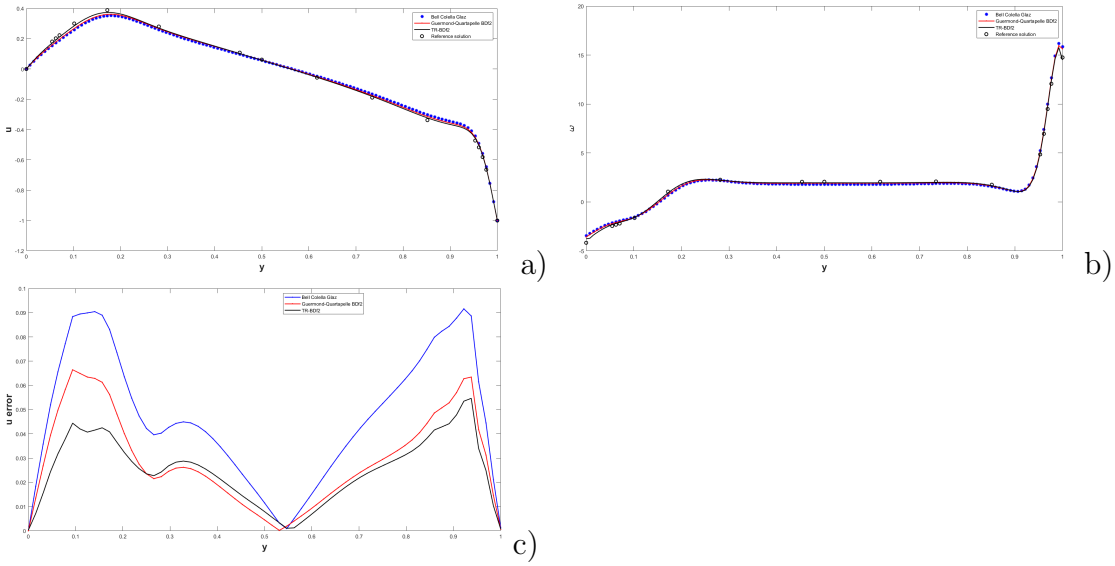


Figure 3: Lid driven cavity benchmark at $Re = 1000$: a) u velocity component values along the middle of the cavity, b) ω values along the middle of the cavity. The continuous lines denote the numerical solutions with the methods [10], [33] and with the present method, the circles the reference solution values from [15]. c) absolute error on u velocity component with respect to reference solution of [15] interpolated along the middle of the cavity. The continuous black line denotes the result with the proposed method, the red one the results of [33] and the blue dots the results of [10].

Moreover, we have compared the computational time required by the three methods for $\mathcal{H} = \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$, keeping the Courant number fixed. This assessment is important to show potential drawbacks of the two stage of the TR-BDF2 method with respect to the single stage methods employed in [10, 33]. As shown in Figure 4, the TR-BDF2 method shows superior efficiency with respect to the Bell-Colella-Glaz method, while it behaves similarly to the BDF2 method of [33]. Multistep methods, however, entail a memory overhead that is not appealing for large scale applications.

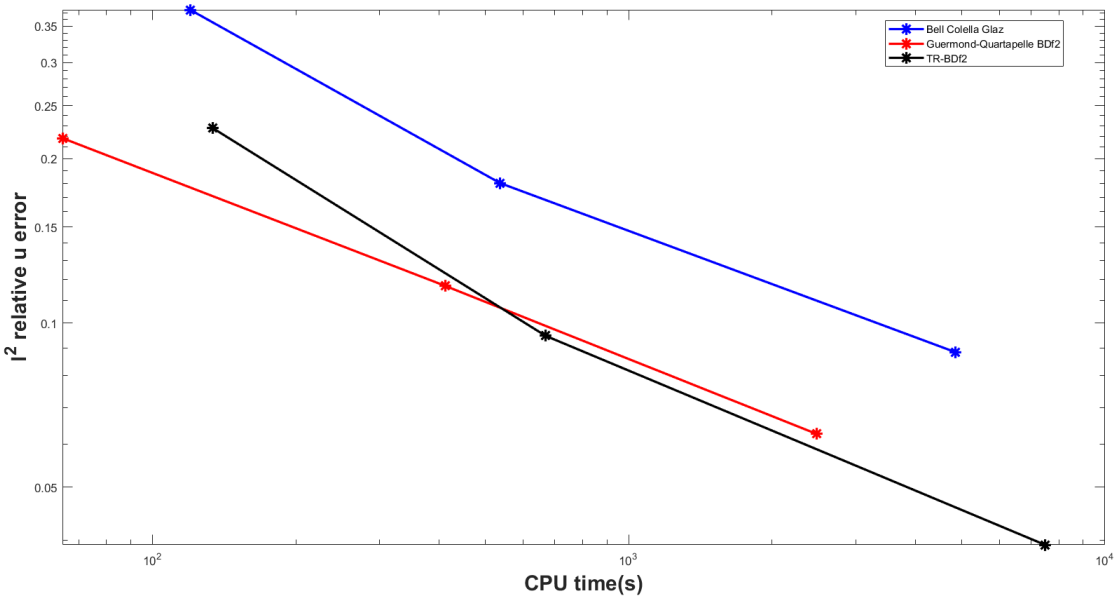


Figure 4: Lid driven cavity benchmark at $Re = 1000$, l^2 relative errors with respect to the reference solution values from [15] interpolated along the middle. The continuous black line denotes the result with the proposed method, the red one the results of [33] and the blue dots the results of [10].

We have also repeated this test using the adaptive tools present in the *deal II* library, as mentioned at the beginning of the Section. In each element K we define the quantity

$$\eta_K = \mathrm{diam}(K)^2 \, \|\boldsymbol{\omega}\|_K^2 \tag{53}$$

that acts as local refinement indicator. We then started from a uniform Cartesian mesh with $N_e = 8$ in each coordinate direction and we allowed refinement or coarsening based on the distribution of the values of η_K, refining 10% of the elements with largest indicator values and coarsening 30% of the elements with the smallest indicator values. This remeshing procedure was carried out every 1000 time steps. However, in order to avoid using too coarse a mesh for too long in the initial stages of the simulation, every 50 time steps the maximum difference between the velocities at two consecutive time steps was checked and the remeshing was performed whenever this quantity was greater then 10^{-2}. The minimum element diameter allowed was $\mathcal{H} = \frac{1}{128}$, so as to obtain again $C \approx 1.3$. A maximum element diameter equal to $\frac{1}{32}$ was also required, in order to avoid an excessive reduction of the spatial resolution. The final adapted mesh and the streamline contours are reported in Figure 5. It can be observed that the refinement indicator allows to enhance automatically the resolution along the top boundary of the domain and in other regions of large vorticity values.

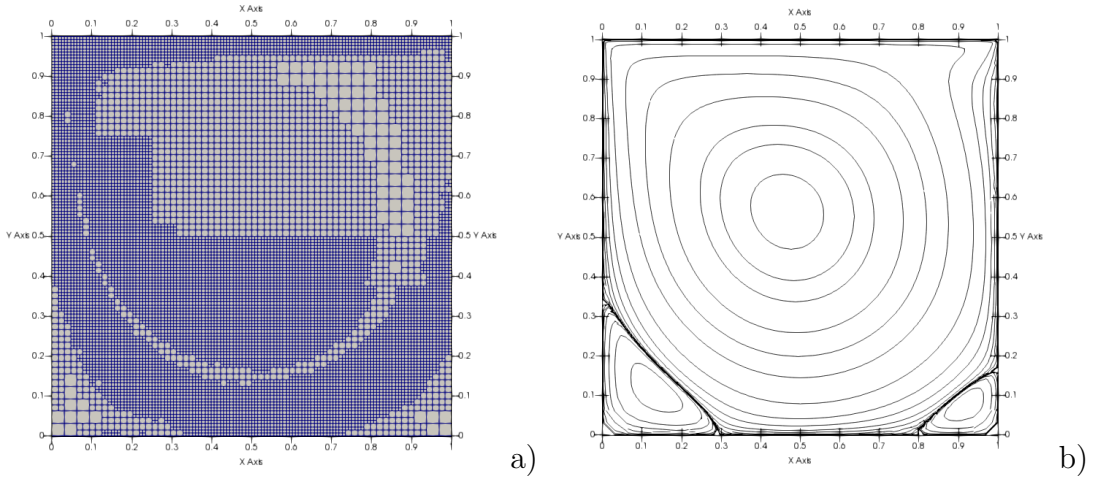


Figure 5: Lid driven cavity benchmark at $Re = 1000$, adaptive simulation: a) final mesh after adaptive refinement, b) streamfunction contours. Contour values as in [16].

For a more quantitative point of view, we compare again in Figure 6 the u component of the velocity and the vorticity ω along the middle of the cavity with the reference results in [15]. The maximum horizontal velocity along the centerline is now $u_{max} = 0.3739$ which implies a relative error of the order of 10^{-2}, as in the corresponding non adaptive simulation. The vorticity value at the center of the cavity is now $\omega_{cen} = 1.9652$, which also implies a relative error with respect to the reference solution of the order of 10^{-2}. In Figure 7, instead, the absolute difference between the velocities computed in the fixed mesh and adaptive simulations is plotted over the whole domain, showing that no substantial loss of accuracy has occurred. This result has been obtained with a reduction of about 25% of the required computational time. While showing the potential of the adaptivity procedures available in the present implementation, this is still far from optimal. Experiments with more specific error indicators and less restrictive options for the refinement parameters will be carried out in future work.

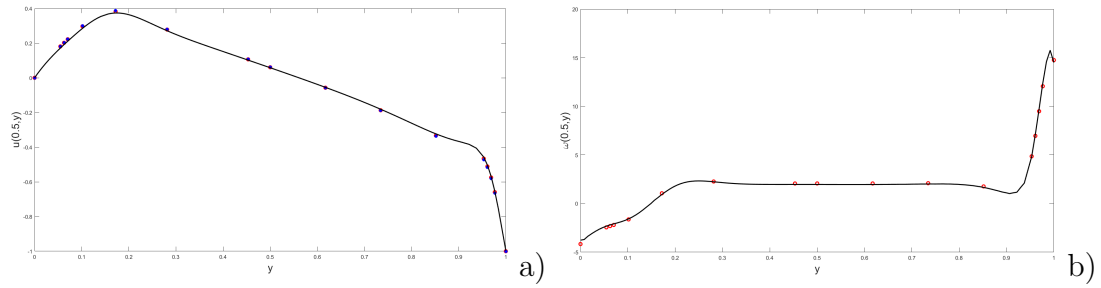


Figure 6: Lid driven cavity benchmark at $Re = 1000$, adaptive simulation: a) u velocity component values along the middle of the cavity, b) ω values along the middle of the cavity. The continuous line denotes the numerical solution and the dots the reference solution values from [15].

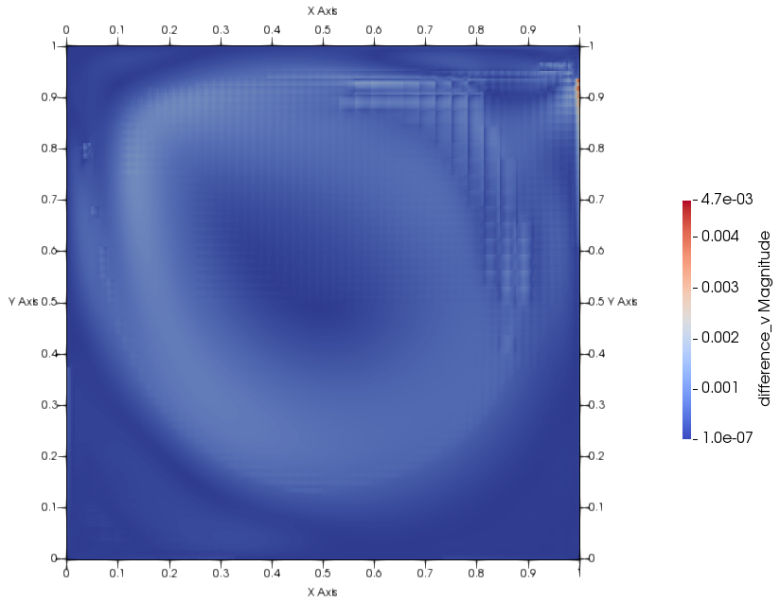


Figure 7: Lid driven cavity benchmark at $Re = 1000$, difference for velocity magnitude between the fixed grid simulation and the adaptive simulation (interpolated to the fixed grid).

4.3 Three-dimensional lid driven cavity

We now consider the three-dimensional analog of the previously studied lid driven cavity benchmark. Among several others, we consider the configuration and reference solutions provided in [1], which we summarize here for convenience. We consider a rectangular cavity of the size $d \times h \times l$ in the

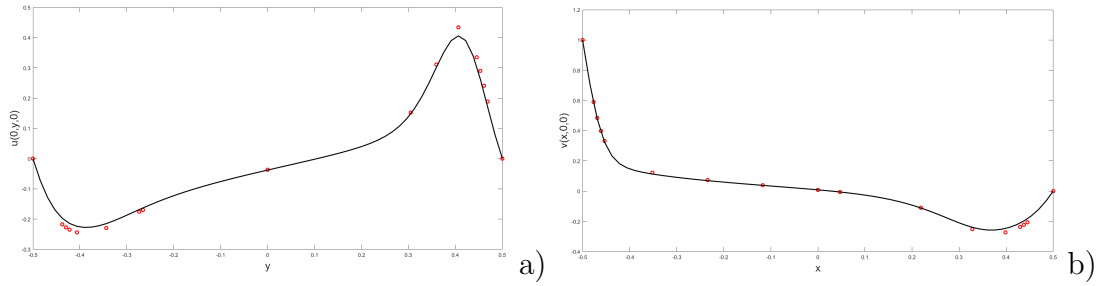
x, y and z direction, respectively. The flow is driven by the wall at $x = d/2$, which moves tangentially in the y direction with constant velocity V. The length d is used to introduce non dimensional space variables, so that the effective computational domain is given by

$$\Omega = \left[-\frac{\Gamma}{2}, \frac{\Gamma}{2}\right] \times \left[-\frac{1}{2}, \frac{1}{2}\right] \times \left[-\frac{\Lambda}{2}, \frac{\Lambda}{2}\right],$$

where the aspect ratios in the x and z directions are defined as

$$\Gamma = \frac{h}{d} \qquad \Lambda = \frac{l}{d}. \tag{54}$$

We have considered here the $\Gamma = 1, \Lambda = 1$ case, computed with $\mathbf{Q}_2 - Q_1$ elements on a Cartesian mesh composed of $64 \times 64 \times 48$ square elements, with a time step chosen so that the Courant number is approximately 1. Notice that the same mesh was employed in [1], which however employed a much more accurate spectral collocation method. The computation is performed until the steady state is reached up to a tolerance of 10^{-4}, which is achieved around $T = 40$. We take as reference results those presented in Tables 5 and 6 in [1]. Notice that, in that paper, a different non dimensional scaling is employed, so that their results have been appropriately rescaled in order to compare them with those obtained here. In Figure 8 we report the results for the v velocity component values along the x axis and the u component of the velocity along the y axis, respectively. We see that, in spite of the relatively coarse mesh, a reasonable accuracy is achieved.



a) b)

Figure 8: 3D lid driven cavity benchmark at $Re = 1000$, fixed mesh simulation with $N_e =$ elements in each coordinate directions, a) v velocity component values along the x axis b) u velocity component values along the y axis. The continuous line denotes the numerical solution and the dots the reference solution values from [1].

In Figure 9 and 10 we show instead the velocity field on the three median plane sections of the cavity, highlighting the presence of vortices near the centerline of the cavity. The results are in good qualitative agreement with those reported in [37].

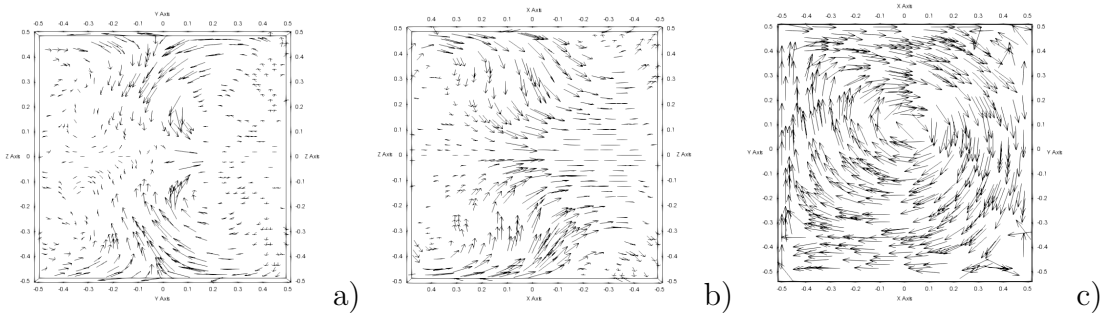


Figure 9: 3D lid driven cavity benchmark at $Re = 1000$, a) Flow field vectors for the plane $x = 0$, b) Flow field vectors for the plane $y = 0$, c) Flow field vectors for the plane $z = 0$.

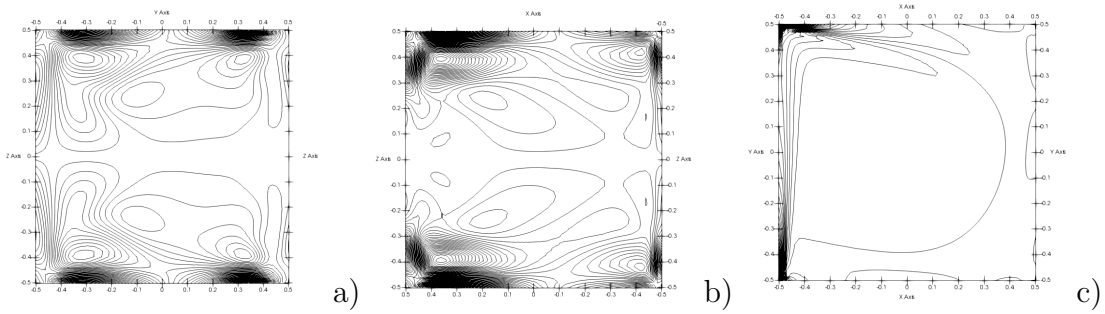
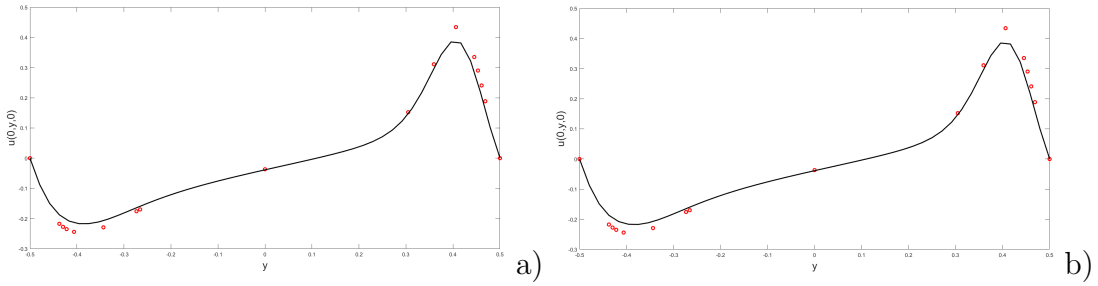


Figure 10: 3D lid dirven cavity benchmark at $Re = 1000$, a) Vorticity (ω_x) contours at $x = 0$, b) Vorticity (ω_y) contours at $y = 0$, Vorticity (ω_z) contours at $z = 0$

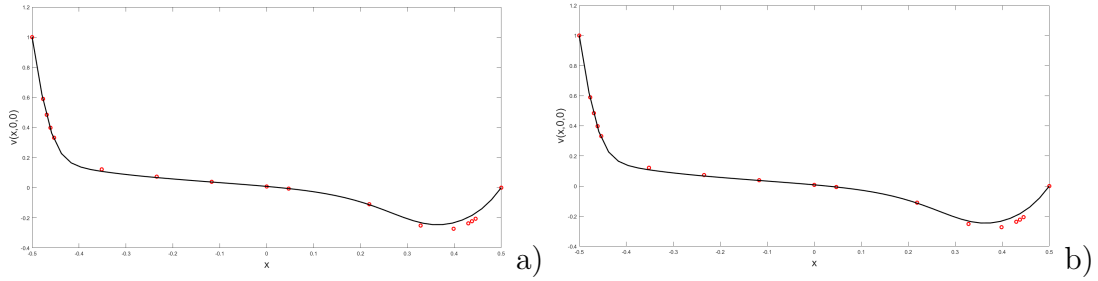
We have also exploited again the mesh adaptivity tool provided by *deal.II* with the same refinement indicator introduced for the two-dimensional test. In particular, we started from a coarse mesh with $N_e = 6$ elements along each direction and again we performed the refinement procedure on at most 10% of the elements with the largest indicator value every 1000 time steps, while coarsening on at most 30% of the elements with the smallest indicator values; moreover we have checked every 50 time steps if the refinement procedure had to be performed in advance in case the maximum difference between the velocities at two consecutive time steps was greater then 10^{-2}. The minimum element diameter allowed was $\mathcal{H} = \frac{1}{48}$ in order to obtain $C \approx 1$. In Figure 11 and in Figure 12 we report again the results for the v velocity component values along the x axis and the u component of the velocity along the y axis, respectively, compared with the results obtained using a fixed grid with $N_e = 48$ elements along each direction. One can notice very good agreement between the two simulations, while the computational time required to perform the adaptive simulation is about half of that required

by the fixed grid simulation. Moreover, we have compared in Figure 13 the errors of the two components for the velocity for the fixed and adaptive mesh, respectively. It is clear that, in spite of the different computational time, no significant differences arise.



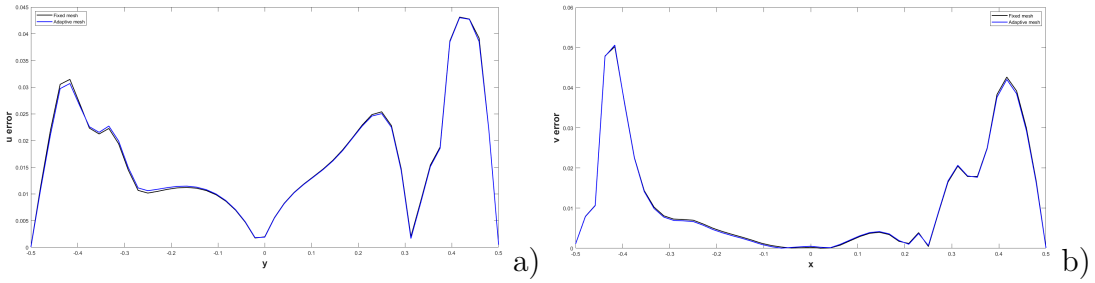
a) b)

Figure 11: 3D lid driven cavity benchmark at $Re = 1000$, a) u velocity component values along the y axis for adaptive mesh simulation, b) u velocity component values along the y axis for fixed grid simulation. The continuous line denotes the numerical solution and the dots the reference solution values from [1].



a) b)

Figure 12: 3D lid driven cavity benchmark at $Re = 1000$, a) v velocity component values along the x axis for adaptive mesh simulation, b) v velocity component values along the x axis for fixed grid simulation. The continuous line denotes the numerical solution and the dots the reference solution values from [1].



a) b)

Figure 13: 3D lid driven cavity benchmark at $Re = 1000$, a) u velocity component component for the plane $x = 0, z = 0$ with reference solution values from [1] interpolated, b) v velocity component comparison for the plane $y = 0, z = 0$ with reference solution values from [1] interpolated. The continuous black line denotes the result with fixed mesh, the blue one denotes the results with adaptive mesh refinement.

5 Conclusions and future perspectives

Building on the experience of [47], we have proposed an accurate, efficient and robust projection method, based on the TR-BDF2 method. While time discretizations of the incompressible Navier-Stokes equations based on accurate implicit solvers have been proposed in many other papers, the specific combination of techniques presented in this work appears to be optimal under several viewpoints for the development of a second order adaptive flow solver.

The proposed fully implicit method has been implemented using discontinuous finite elements in the framework of the numerical library *deal.II*, with the aim of building a reliable, flexible and easily accessible tool for industrial applications that can ultimately be competitive with more conventional finite volume techniques. We have shown that the method has superior accuracy and efficiency with respect to some well known alternative schemes on a number of classical benchmarks.

In future work, besides application of the proposed approach to significant industrial applications and extensions to fully compressible and multiphase flow, an interesting development will be represented by the integration of more sophisticated *a posteriori* error estimation techniques [27, 34, 42] to obtain optimal adaptive approaches. Furthermore, the multirate time integration version of the TR-BDF2 method [11] could also be integrated in the discretization approach, so as to obtain a fully space-time adaptive technique based on a robust and unconditionally stable method.

Acknowledgements

L.B. would like to thank Roberto Ferretti, Elisabetta Carlini and M. Gómez Mármol for several useful discussions.

References

[1] S. Albensoeder and H.C. Kuhlmann. Accurate three-dimensional lid-driven cavity flow. *Journal of Computational Physics*, 206:536–558, 2005.

[2] D.N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM Journal of Numerical Analysis*, 19:742–760, 1982.

[3] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of Discontinuous Galerkin methods for elliptic problems. *SIAM Journal of Numerical Analysis*, 39:1749–1779, 2002.

[4] F. Auteri, N. Parolini, and L. Quartapelle. Numerical investigation on the stability of singular driven cavity flow. *Journal of Computational Physics*, 183:1–25, 2002.

[5] W. Bangerth, R. Hartmann, and G. Kanschat. deal II: a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software (TOMS)*, 33:24–51, 2007.

[6] R.E. Bank, W.M. Coughran, W. Fichtner, E.H. Grosse, D.J. Rose, and R.K. Smith. Transient Simulation of Silicon Devices and Circuits. *IEEE Transactions on Electron Devices.*, 32:1992–2007, 1985.

[7] F. Bassi, A. Crivellini, D.A. Di Pietro, and S. Rebay. An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows. *Computers & Fluids*, 36:1529–1546, 2007.

[8] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous galerkin solution of the Reynolds-averaged Navier–Stokes and k–ω turbulence model equations. *Computers & Fluids*, 34:507–540, 2005.

[9] F. Bassi, L.Botti, A. Colombo, A. Ghidoni, and F. Massa. Linearly implicit Rosenbrock-type Runge–Kutta schemes applied to the Discontinuous Galerkin solution of compressible and incompressible unsteady flows. *Computers & Fluids*, 118, 2015.

[10] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressibleNavier-Stokes equations. *Journal of Computational Physics*, 85:257–283, 1989.

[11] L. Bonaventura, F. Casella, L. Delpopolo Carciopolo, and A. Ranade. A self adjusting multirate algorithm for robust time discretization of

partial differential equations. *Computers and Mathematics with Applications*, 79:2086–2098, 2020.

[12] L. Bonaventura, E. D. Fernández-Nieto, J. Garres-Díaz, and G. Narbona-Reina. Multilayer shallow water models with locally variable number of layers and semi-implicit time discretization. *Journal of Computational Physics*, 364:209–234, 2018.

[13] L. Bonaventura and M. Gómez Marmol. The TR-BDF method for second order problems in structural mechanics. *Computers & Mathematics with Applications*, 95:13–26, 2021.

[14] L. Bonaventura and A. Della Rocca. Unconditionally strong stability preserving extensions of the TR-BDF2 method. *Journal of Scientific Computing*, 70:859–895, 2017.

[15] O. Botella and R. Peyret. Benchmark spectral results on the lid-driven cavity flow. *Computers and Fluids*, 27:421–433, 1998.

[16] C.H. Bruneau and M. Saad. The 2D lid-driven cavity flow revisited. *Computers and Fluids*, 35:326–348, 2006.

[17] G. Chen, Q. Xiong, P.J. Morris, E.G. Paterson, A. Sergeev, and Y. Wang. OpenFOAM for computational fluid dynamics. *Notices of the AMS*, 61:354–363, 2014.

[18] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2:12–26, 1967.

[19] A.J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22:745–762, 1968.

[20] J.A. Ekaterinaris. High-order accurate numerical solutions of incompressible flows with the artificial compressibility method. *International Journal of Numerical Methods in Fluids*, 45:1187–1207, 2004.

[21] N. Fehn, M. Kronbichler, C. Lehrenfeld, G. Lube, and P. Schroeder. High-order DG solvers for under-resolved turbulent incompressible flows: A comparison of l^2 and $h(div)$ methods. *International Journal of Numerical Methods in Fluids*, 91:533–556, 2019.

[22] N. Fehn, W.A. Wall, and M. Kronbichler. On the stability of projection methods for the incompressible Navier–Stokes equations based on high-order discontinuous Galerkin discretizations. *Journal of Computational Physics*, 351:392–421, 2017.

[23] N. Fehn, W.A. Wall, and M. Kronbichler. Robust and efficient discontinuous G alerkin methods for under-resolved turbulent incompressible flows. *Journal of Computational Physics*, 372:667–693, 2018.

[24] C. A. J. Fletcher. *Computational techniques for fluid dynamics, Volume 1: Fundamental and general techniques.* Springer Verlag, 1997.

[25] D. Galloway and U. Frisch. A note on the stability of a family of space-periodic Beltrami flows. *Journal of Fluid Mechanics*, 158:557–564, 1987.

[26] J. Garres-Díaz and L. Bonaventura. Flexible and efficient discretizations of multilayer models with variable density. *Applied Mathematics and Computation*, 402:126097, 2021.

[27] E.H. Georgoulis, E. Hall, and P. Houston. Discontinuous Galerkin methods on *hp*-anisotropic meshes II: A posteriori error analysis and adaptivity. *Applied Numerical Mathematics*, 59:2179–2194, 2009.

[28] G. Giorgiani, S. Fernández-Méndez, and A. Huerta. Hybridizable discontinuous Galerkin with degree adaptivity for the incompressible Navier–Stokes equations. *Computers & Fluids*, 98:196–208, 2014.

[29] F.X. Giraldo. *An Introduction to Element-Based Galerkin Methods on Tensor-Product Bases.* Springer Nature, 2020.

[30] F.X. Giraldo, J.F. Kelly, and E.M. Constantinescu. Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (NUMA). *SIAM Journal of Scientific Computing*, 35:1162–1194, 2013.

[31] A. Green and G.I. Taylor. Mechanism of the production of small eddies from large ones. *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, 158:499–521, 1937.

[32] J.L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Computer methods in applied mechanics and engineering*, 195:6011–6045, 2006.

[33] J.L. Guermond and L. Quartapelle. On incremental projection methods. In *International Conference on Navier-Stokes Equations: Theory and Numerical Methods*, volume 338, pages 277–288. Pitman, 1998.

[34] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: Goal–oriented a posteriori error estimation. *International Journal of Numerical Analysis & Modeling*, 3:141–162, 2006.

[35] M.E. Hosea and L.F. Shampine. Analysis and implementation of TR-BDF2. *Applied Numerical Mathematics*, 20:21–37, 1996.

[36] H. Jasak, A. Jemcov, Zeljko Tukovic, et al. OpenFOAM: A C++ library for complex physics simulations. In *International Workshop on Coupled Methods in Numerical Dynamics*, volume 1000, pages 1–20. IUC Dubrovnik Croatia, 2007.

[37] B. Jiang, T.L. Lin, and L. A. Povinelli. Large-scale computation of incompressible viscous flow by least-squares finite element method. *Com-*

puter Methods in Applied Mechanics and Engineering, 114(3):213–231, 1994.

[38] G.E. Karniadakis and S. Sherwin. *Spectral hp−Element Methods for Computational Fluid Dynamics*. Oxford University Press, 2005.

[39] C.A. Kennedy and M.H. Carpenter. Diagonally implicit Runge-Kutta methods for ordinary differential equations, a review. Technical Report TM-2016-219173, NASA, 2016.

[40] C. Merkle. Time-accurate unsteady incompressible flow algorithms based on artificial compressibility. In *8th Computational Fluid Dynamics Conference*, page 1137, 1987.

[41] P. Nithiarasu. An efficient artificial compressibility (AC) scheme based on the characteristic based split (CBS) method for incompressible flows. *International Journal for Numerical Methods in Engineering*, 56:1815–1845, 2003.

[42] T. J. Oden, W. Wu, and M. Ainsworth. An a posteriori error estimate for finite element approximations of the Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 111:185–202, 1994.

[43] L. Quartapelle. *Numerical solution of the incompressible Navier-Stokes equations*. Birkhäuser, 2013.

[44] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*, volume 23. Springer Science & Business Media, 2008.

[45] M.M. Rahman and T. Siikonen. An artificial compressibility method for viscous incompressible and low Mach number flows. *International Journal for Numerical Methods in Engineering*, 75:1320–1340, 2008.

[46] S. Rhebergen, B. Cockburn, and J.J.W. Van Der Vegt. A space–time discontinuous galerkin method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 233:339–358, 2013.

[47] A. Della Rocca. *Large-Eddy Simulations of Turbulent Reacting Flows with Industrial Applications*. PhD thesis, Politecnico di Milano, 2 2018.

[48] S.E. Rogers and D. Kwak. Upwind differencing scheme for the time-accurate incompressible Navier-Stokes equations. *AIAA journal*, 28:253–262, 1990.

[49] D. Schötzau, C. Schwab, and A. Toselli. Stabilized DGFEM for incompressible flows. *Mathematical Models and Methods in Applied Sciences*, 13:1413–1436, 2003.

[50] M. Tavelli and M. Dumbser. A staggered semi-implicit discontinuous Galerkin method for the two dimensional incompressible Navier–Stokes equations. *Applied Mathematics and Computation*, 248:70–92, 2014.

[51] M. Tavelli and M. Dumbser. A staggered space–time discontinuous Galerkin method for the three-dimensional incompressible Navier–Stokes equations on unstructured tetrahedral meshes. *Journal of Computational Physics*, 319:294–323, 2016.

[52] R. Temam. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (ii). *Archive for Rational Mechanics and Analysis*, 33:377–385, 1969.

[53] A. Toselli. $h - p$ Discontinuous Galerkin approximations for the Stokes problem. *Mathematical Models and Methods in Applied Sciences*, 12:1565–1597, 2002.

[54] G. Tumolo and L. Bonaventura. A semi-implicit, semi-Lagrangian discontinuous Galerkin framework for adaptive numerical weather prediction: SISL-Dg framework for adaptive NWP. *Quarterly Journal of the Royal Meteorological Society*, 141:2582–2601, 2015.

[55] G. Tumolo, L. Bonaventura, and M. Restelli. A semi-implicit, semi-Lagrangian, $p-$adaptive discontinuous Galerkin method for the shallow water equations. *Journal of Computational Physics*, 232:46–67, 2013.

[56] H.G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12:620–631, 1998.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

45/2021 Diquigiovanni, J.; Fontana, M.; Vantini, S.
Distribution-Free Prediction Bands for Multivariate Functional Time Series: an Application to the Italian Gas Market

46/2021 Diquigiovanni, J.; Fontana, M.; Vantini, F.
Conformal Prediction Bandsfor Multivariate Functional Data

44/2021 Gentili, G.G.; Khosronejad, M.; Bernasconi, G.; Perotto, S.; Micheletti, S.
Efficient Modeling of Multimode Guided Acoustic Wave Propagation in Deformed Pipelines by Hierarchical Model Reduction

43/2021 Salvador, M.; Fedele, M.; Africa, P.C.; Sung, E.; Dede', L.; Prakosa, A.; Chrispin, J.; Trayanov
Electromechanical modeling of human ventricles with ischemic cardiomyopathy: numerical simulations in sinus rhythm and under arrhythmia

42/2021 Calissano, A.; Fontana, M.; Zeni, G.; Vantini, S.
Conformal Prediction Sets for Populations of Graphs

41/2021 Costa, G., Cavinato, L., Maschi, C., Fiz, F., Sollini, M., Politi, L. S., Chiti, A., Balzarini, L., Ag
Virtual Biopsy for Diagnosis of Chemotherapy-Associated Liver Injuries and Steatohepatitis: A Combined Radiomic and Clinical Model in Patients with Colorectal Liver Metastases

40/2021 Martinolli, M.; Cornat, F.; Vergara, C.
Computational Fluid-Structure Interaction Study of a new Wave Membrane Blood Pump

39/2021 Barnafi, N.; Di Gregorio, S.; Dede', L.; Zunino, P.; Vergara, C.; Quarteroni, A.
A multiscale poromechanics model integratingmyocardial perfusion and systemic circulation

38/2021 Giusteri, G. G.; Miglio, E.; Parolini, N.; Penati, M.; Zambetti, R.
Simulation of viscoelastic Cosserat rods based on the geometrically exact dynamics of special Euclidean strands

37/2021 Dassi, F.; Fumagalli, A.; Mazzieri, I.; Scotti, A.; Vacca, G.
A Virtual Element Method for the wave equation on curved edges in two dimensions