sciendo

CAIM
ISSN 2038-0909

Commun. Appl. Ind. Math. 13 (1), 2022, 21–38 DOI: 10.2478/caim-2022-0003

Research Article

High order Finite Difference/Discontinuous Galerkin schemes for the incompressible Navier-Stokes equations with implicit viscosity

Walter Boscheri[1]*, Maurizio Tavelli[2], Nicola Paoluzzi[1]

[1]Department of Mathematics and Computer Science, University of Ferrara, Ferrara, Italy

[2]Faculty of Computer Science, University of Bozen, Bolzano, Italy

*Email address for correspondence: walter.boscheri@unife.it

Communicated by Luca Formaggia

Received on 04 12, 2022. Accepted on 05 31, 2022.

Abstract

In this work we propose a novel numerical method for the solution of the incompressible Navier-Stokes equations on Cartesian meshes in 3D. The semi-discrete scheme is based on an explicit discretization of the nonlinear convective flux tensor and an implicit treatment of the pressure gradient and viscous terms. In this way, the momentum equation is formally substituted into the divergence-free constraint, thus obtaining an elliptic equation on the pressure which eventually maintains at the discrete level the involution on the divergence of the velocity field imposed by the governing equations. This makes our method belonging to the class of so-called structure-preserving schemes. High order of accuracy in space is achieved using an efficient CWENO reconstruction operator that is exploited to devise a conservative finite difference scheme for the convective terms. Implicit central finite differences are used to remove the numerical dissipation in the pressure gradient discretization. To avoid the severe time step limitation induced by the viscous eigenvalues related to the parabolic terms in the governing equations, we propose to devise an implicit local discontinuous Galerkin (DG) solver. The resulting viscous sub-system is symmetric and positive definite, therefore it can be efficiently solved at the aid of a matrix-free conjugate gradient method. High order in time is granted by a semi-implicit IMEX time stepping technique. Convergence rates up to third order of accuracy in space and time are proven, and a suite of academic benchmarks is shown in order to demonstrate the robustness and the validity of the novel schemes, especially in the context of high viscosity coefficients.

Keywords: `incompressible Navier-Stokes, IMEX schemes, implicit viscosity terms, high order in space and time`

AMS subject classification: `65Mxx, 65Yxx`

1. Introduction

The incompressible Navier-Stokes equations constitute a well-established physical model which governs fluids in the incompressible limit that is obtained from the compressible Navier-Stokes equations. Indeed, in the low Mach limit the pressure of compressible gases tends to become constant and therefore an involution related to the divergence of the velocity field is retrieved from the energy equation. From the numerical viewpoint, the main difficulty is given by the solution of the pressure Poisson equation such that the divergence-free constraint is maintained at the discrete level. The satisfaction of the divergence-free condition at a local discrete level, i.e. within each computational cell, plays a crucial role when dealing with transports of solutes or other quantities by the main flow field. The computation of the pressure which respects the divergence-free involution typically involves an associated linear equation system to be solved, as originally proposed within the SIMPLE method [1].

The incompressible Navier-Stokes equations take the form of a saddle-point problem than can be formulated by means of a set of elliptic problems, thus boundary conditions affect instantaneously the solution everywhere inside the domain. As a consequence, continuous finite element methods are very popular [2–8] as well as finite difference schemes [9–12]. Discontinuous Galerkin methods for the incompressible Navier-Stokes equations have been recently investigated in [13–20]. The technique of artificial compressibility, originally introduced by Chorin in [21], has been used in the DG framework [22]. Staggered grids combined with finite difference schemes and semi-implicit time discretizations have been widely studied in the literature [9–12,23–25] to devise structure-preserving methods that can respect the divergence-free involution on the velocity field.

A similar, though different, time stepping technique is given by the family of implicit-explicit (IMEX) methods [26–30] which have recently been employed to design asymptotic preserving methods for the inviscid [31] and viscous [32,33] compressible flows. IMEX-BDF schemes have been investigated in [34,35], while IMEX combined with continuous finite element schemes have been presented in [36].

In this work we aim at developing a high order numerical scheme in space and time for the full 3D incompressible Navier-Stokes equations on cell-centered Cartesian grids. The nonlinear convective terms will be discretized explicitly using a finite difference conservative formulation that is based on an efficient CWENO reconstruction technique [37,38]. The pressure is taken implicitly, so that the momentum equation is inserted into the divergence-free constraint to obtain an elliptic Poisson equation on the pressure that guarantees the preservation of the involution up to machine precision. An additional novelty is given by the implicit discretization of the viscous terms, relying on a local weak formulation of the viscous sub-system. The idea comes from the semi-implicit discontinuous Galerkin methods proposed in [39], and it is here extended to a complete different setup given by cell-centered finite difference schemes with Runge-Kutta time marching.

The outline of this article is as follows. In Section 2 we introduce the governing equations, while Section 3 is devoted to the description of the numerical method. Specific care is taken to detail the solution of the Poisson equation on the pressure and the novel implicit discretization of the viscous subsystem. Numerical convergence studies and applications to a set of test problems is shown in Section 4. A concluding section finalizes the article where we draw some conclusions and present an outlook to future research.

2. Governing equations

The computational domain $\Omega \in \mathbb{R}^d$ is defined in space dimension $d \in \{1, 2, 3\}$ with the vector of spatial coordinates $\boldsymbol{x} = (x, y, z) \in \Omega$ and time variable $t \in \mathbb{R}^+$. The incompressible Navier-Stokes equations are given by

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \mathbf{F}_c + \nabla p - \nu \nabla^2 \boldsymbol{u} = \mathbf{0}, \tag{1a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1b}$$

where $\boldsymbol{u} = (u, v, w)$ is the velocity vector with components along each spatial direction and $p = P/\rho$ is the normalized fluid pressure, P being the physical pressure and ρ denoting the constant fluid density. The kinematic viscosity coefficient is computed by $\nu = \mu/\rho$, and μ is the dynamic viscosity of the fluid. The flux tensor of the nonlinear convective terms explicitly writes

$$\mathbf{F}_c := \boldsymbol{u} \otimes \boldsymbol{u} = \begin{bmatrix} uu & uv & uw \\ vu & vv & vw \\ wu & wv & ww \end{bmatrix}. \tag{2}$$

In what follows we will present a numerical discretization for the model (1).

3. Numerical scheme

The time coordinate is bounded in the interval $t \in [0; t_f]$, and the final time t_f is reached performing a sequence of time steps $\Delta t = t^{n+1} - t^n$. The size of the time step is determined according to a CFL-type stability condition based on the magnitude of the flow velocity, that is

$$\Delta t \leq \text{CFL} \min_{\Omega} \left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} \right)^{-1}, \qquad \text{CFL} \in (0; 1). \tag{3}$$

Notice that there is no dependency on the parabolic viscous terms because they will be discretized implicitly, hence remarkably improving the efficiency of the resulting scheme.

The computational domain $\mathbf{\Omega}(\boldsymbol{x}) = [x_{\min}; x_{\max}] \times [y_{\min}; y_{\max}] \times [z_{\min}; z_{\max}]$ is discretized using a Cartesian grid composed of a total number $N_e = N_x \times N_y \times N_z$ of cells $C_{i,j,k}$ with volume $|C_{i,j,k}| = \Delta x \, \Delta y \, \Delta z$. The characteristic mesh sizes are given by

$$\Delta x = \frac{x_{\max} - x_{\min}}{N_x}, \qquad \Delta y = \frac{y_{\max} - y_{\min}}{N_y}, \qquad \Delta z = \frac{z_{\max} - z_{\min}}{N_z}, \tag{4}$$

while the cell center is indicated as $x_C = (x_i, y_j, z_k)$ and interfaces are given by half indexes, e.g. $x_{i+1/2} = x_i + \Delta x/2$ is the right interface in $x-$direction of cell $C_{i,j,k}$. The discrete representation of a generic quantity $q(\boldsymbol{x}, t)$ in a given cell $C_{i,j,k}$ at a given time level t^n is labeled with $q^n_{i,j,k}$. Data are stored as point values in the finite difference framework, hence

$$q(\boldsymbol{x}, t^n) = q(\boldsymbol{x}_{i,j,k}, t^n) := q^n_{i,j,k}. \tag{5}$$

Let us also introduce the reference element $C_E = [0; 1]^3$ defined in the reference space with coordinates $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ through the mapping

$$\xi = \xi(x, i) = \frac{1}{\Delta x}\left(x - x_{i-1/2}\right), \quad \eta = \eta(y, j) = \frac{1}{\Delta y}\left(y - y_{j-1/2}\right), \quad \zeta = \zeta(z, k) = \frac{1}{\Delta z}\left(z - z_{k-1/2}\right). \tag{6}$$

The one-dimensional reference interval $I = [0; 1]$ is used to define a nodal basis given by $N + 1$ linearly independent Lagrange interpolating polynomials of maximum degree N, i.e. $\{\phi_l\}_{l=1}^{N+1}$, passing through a set of $N + 1$ nodal points $\{\xi_r\}_{r=1}^{N+1}$, which are assumed to be the Gauss-Lobatto nodes. The interpolation property holds by construction, thus

$$\phi_l(\xi_r) = \delta_{lr}, \qquad l, r = 1, \ldots, N + 1, \tag{7}$$

and the basis can be extended to 3D with a tensor product in all spatial dimensions, that is

$$\phi_m(\xi, \eta, \zeta) = \phi_{l_1}(\xi)\phi_{l_2}(\eta)\phi_{l_3}(\zeta), \qquad l_1, l_2, l_3 = 1, \ldots, N + 1 \qquad m = 1, \ldots, (N + 1)^3, \tag{8}$$

with $m = m(l_1, l_2, l_3)$ being a multi-index. All the integrals appearing in the numerical scheme are evaluated using Gaussian quadrature formulae of suitable accuracy [40].

The set of basis functions (8) is employed for achieving high order of accuracy in space. Starting from the known cell values given by (5), the CWENO reconstruction procedure introduced in [37] permits to compute a high order polynomial representation $w(\boldsymbol{x}, t)$ that writes

$$w(\boldsymbol{x}, t^n) = \phi_m(\xi, \eta, \zeta)\, \hat{w}^n_{ijk,m}, \tag{9}$$

where $\hat{w}^n_{ijk,m}$ denote the expansion coefficients. Einstein summation convention implying summation over repeated indexes is adopted in this article. The reconstruction operator is compactly indicated with $w = \mathbb{R}(q)$, meaning that a quantity q is reconstructed by means of a polynomial w of degree N of the form (9) following the same algorithm detailed in [37].

3.1. *Semi-implicit cell-centered scheme*

The semi-discrete scheme is derived from the following considerations. To avoid severe limitations on the maximum admissible time step, the viscous terms $\nu \, \nabla^2 \boldsymbol{u}$ are discretized implicitly. On the other hand, an explicit treatment of the convective flux tensor $\mathbf{F}_c$ is adopted to remove any non-linearity in the implicit solver. Finally, to satisfy the divergence-free constraint (1b) at the discrete level, the pressure is taken implicitly in (1a) and then the momentum equation is substituted into the continuity equation [23,41] using the fully discrete space-time scheme. In this way, an elliptic equation on the pressure is derived, which guarantees that the new pressure respects the involution on the divergence of the velocity field at the new time level.

Along the lines of the asymptotic preserving methods recently developed for compressible flows [31, 32,42], a semi-implicit IMEX scheme [37,43] can be formulated for the incompressible Navier-Stokes equations (1):

$$\frac{\boldsymbol{u}^{n+1}_{i,j,k} - \boldsymbol{u}^{n}_{i,j,k}}{\Delta t} + \mathbb{F}_{i,j,k}(\boldsymbol{u}^n) + \mathbb{G}_{i,j,k}(p^{n+1}) - \nu\, \mathbb{L}_{i,j,k}(\boldsymbol{u}^{n+1}) = \boldsymbol{0}, \tag{10a}$$

$$\mathbb{D}_{i,j,k}(\boldsymbol{u}^{n+1}) = 0. \tag{10b}$$

The spatial operators $\mathbb{F}_{i,j,k}$, $\mathbb{G}_{i,j,k}$ and $\mathbb{D}_{i,j,k}$ are introduced hereafter referring to a generic variable $q(\boldsymbol{x},t)$ at a given time $t = t^n$, while the discretization of the Laplace operator $\mathbb{L}_{i,j,k}$ will be presented in the next section.

- Numerical flux operator $\mathbb{F}(q)$ in the $x-$direction:

$$\mathbb{F}_i(q) = \frac{1}{\Delta x}\left(\int\limits_{y_{j-1/2}}^{y_{j+1/2}} \int\limits_{z_{k-1/2}}^{z_{k+1/2}} \mathcal{F}_{i+1/2,j,k}(q)\, \Delta y \Delta z - \int\limits_{y_{j-1/2}}^{y_{j+1/2}} \int\limits_{z_{k-1/2}}^{z_{k+1/2}} \mathcal{F}_{i-1/2,j,k}(q)\, \Delta y \Delta z \right). \tag{11}$$

  The numerical flux function $\mathcal{F}(q)$ is evaluated according to [44] using a finite difference formulation

$$\mathcal{F}_{i+1/2,j,k}(q) = \mathbb{R}(f^{+}_{i,j,k}) + \mathbb{R}(f^{-}_{i+1,j,k}), \qquad f^{\pm}_{i,j,k} = \frac{1}{2}(f(q_{i,j,k}) \pm |u_{i,j,k}|\, q_{i,j,k}), \tag{12}$$

  which corresponds to a Lax-Friedrichs flux splitting with $f(\cdot)$ representing the physical flux related to variable q. High order of accuracy is efficiently obtained by: i) performing a CWENO reconstruction for all $f^{\pm}_{i,j,k}$, i.e. $\mathbb{R}(f^{\pm})$ that allows boundary extrapolated numerical fluxes to be obtained, and ii) by using the conservative flux formulation (11) with Gaussian quadrature formulae for the evaluation of the integrals. Notice that this is different from standard high order WENO finite difference schemes [44], since a polynomial (and not a pointwise) representation of the split fluxes $f^{\pm}_{i,j,k}$ is evaluated. The same applies to the other spatial directions, thus $\mathbb{F}_{i,j,k}(q) = \mathbb{F}_i(q) + \mathbb{F}_j(q) + \mathbb{F}_k(q)$. The CWENO reconstruction procedure is fully described in [37].
- The gradient operator $\mathbb{G}(q)$ is based on a central finite difference discretization:

$$\mathbb{G}_{i,j,k}(q) = \begin{pmatrix} \frac{-q_{i+2,j,k}+8q_{i+1,j,k}-8q_{i-1,j,k}+q_{i-2,j,k}}{12\Delta x} + \mathcal{O}(\Delta x^4) \\ \frac{-q_{i,j+2,k}+8q_{i,j+1,k}-8q_{i,j-1,k}+q_{i,j-2,k}}{12\Delta y} + \mathcal{O}(\Delta y^4) \\ \frac{-q_{i,j,k+2}+8q_{i,j,k+1}-8q_{i,j,k-1}+q_{i,j,k-2}}{12\Delta z} + \mathcal{O}(\Delta z^4) \end{pmatrix}. \tag{13}$$

- The divergence operator directly follows from the gradient operator, thus obtaining

$$\begin{aligned} \mathbb{D}_{i,j,k}(q) = {} & \frac{-q_{i+2,j,k} + 8q_{i+1,j,k} - 8q_{i-1,j,k} + q_{i-2,j,k}}{12\Delta x} \\ & + \frac{-q_{i,j+2,k} + 8q_{i,j+1,k} - 8q_{i,j-1,k} + q_{i,j-2,k}}{12\Delta y} \\ & + \frac{-q_{i,j,k+2} + 8q_{i,j,k+1} - 8q_{i,j,k-1} + q_{i,j,k-2}}{12\Delta z} + \mathcal{O}(\Delta x^4, \Delta y^4, \Delta z^4). \end{aligned} \tag{14}$$

The semi-implicit scheme (10) is solved as follows. The Laplace operator related to the viscous terms in the momentum equation (10a) is evaluated implicitly, hence obtaining a provisional velocity field $\boldsymbol{u}^{*}_{i,j,k}$

at the new time level which already contains the computation of both explicit convection and implicit viscous contribution, that is given by the solution of the following equation:

$$\boldsymbol{u}^*_{i,j,k} - \Delta t\, \nu\, \mathbb{L}_{i,j,k}(\boldsymbol{u}^*) = \boldsymbol{u}^n_{i,j,k} - \Delta t\, \mathbb{F}_{i,j,k}(\boldsymbol{u}^n) - \Delta t\, \mathbb{G}_{i,j,k}(p^n). \tag{15}$$

Due to the symmetry and positive definiteness of the Laplace operator that will be introduced in the next section, the system (15) can be efficiently solved using a matrix-free conjugate gradient method. Notice that the above linear system involves all components of the velocity vector, which can however be solved in a decoupled manner thanks to the component-wise definition of the viscous forces. This is different from the compressible Navier-Stokes system [32,42], where the viscous stress tensor fully couples all the components of the gradient of the velocity vector, thus requiring more computational efforts to solve the viscous sub-system (15). Once the solution is found, the momentum equation (10a) reads

$$\frac{\boldsymbol{u}^{n+1}_{i,j,k} - \boldsymbol{u}^*_{i,j,k}}{\Delta t} + \mathbb{G}_{i,j,k}(p^{n+1}) - \mathbb{G}_{i,j,k}(p^n) = \mathbf{0}. \tag{16}$$

The above equation is substituted in its fully discrete formulation into the fully discrete divergence-free constraint (10b), hence obtaining an elliptic equation on the pressure $p^{n+1}_{i,j,k}$:

$$\mathbb{D}_{i,j,k}\left(\boldsymbol{u}^*_{i,j,k} + \Delta t\, \mathbb{G}_{i,j,k}(p^n) - \Delta t\, \mathbb{G}_{i,j,k}(p^{n+1})\right) = 0. \tag{17}$$

The new pressure p^{n+1} does thus guarantee that the divergence-free constraint is satisfied up to the precision of the solution of the associated linear system. Typically we prescribe a tolerance $\varepsilon = 10^{-12}$ while solving the system (17) with the GMRES method [45].

3.2. *High order implicit Laplace operator*

The Laplace operator $\mathbb{L}_{i,j,k}$ in the semi-implicit scheme (10) will be solved at the aid of a discontinuous Galerkin (DG) discretization as done in [39]. Notice that the reconstruction polynomial $w(\boldsymbol{x},t)$ obtained using the operator $\mathbb{R}$ can be directly interpreted as a piecewise DG representation of the quantity $q(\boldsymbol{x},t)$. In this section the upper indexes (p,q,r) indicate the element of the mesh, while lower indexes are used for the degrees of freedom of the DG expansion according to (9). Therefore we label with $\mathbb{L}^{p,q,r}$ the same object addressed with $\mathbb{L}_{i,j,k}$ in the discretization (10).

Let us assume to start from a generic cell-centered scalar field $v^{p,q,r}(\boldsymbol{x}) \approx \phi_k(\boldsymbol{\xi}(\boldsymbol{x}, C^{p,q,r}))\hat{v}^{p,q,r}_k$ for all elements $C^{p,q,r}$. The stencil needed to derive the final Laplace operator is shown in Figure 1 for the two-dimensional case. The construction of $\mathbb{L}^{p,q,r}$ is done in two steps: i) evaluation of a face-staggered gradient, and ii) evaluation of a cell-centered divergence operator that is based on the previously computed face-based gradients. Since the viscous terms in the momentum equations are given as the Laplacian of the velocity field along each single momentum direction, the gradient and divergence operators can be efficiently defined in a decoupled manner performing an efficient direction-by-direction approach. Therefore, let us consider the one-dimensional case along the $x-$direction. We aim at evaluating the face-staggered gradient $\alpha^{p+\frac{1}{2}}$ given by

$$\alpha^{p+\frac{1}{2}} = \frac{\partial v(\boldsymbol{x})}{\partial x}, \qquad \boldsymbol{x} \in [x^p; x^{p+1}]. \tag{18}$$

A variational formulation of (18) over the face-staggered control volume yields

$$\begin{aligned}\int\limits_{x^p}^{x^{p+1}} \phi_k \phi_l \hat{\alpha}_l{}^{p+\frac{1}{2}}\, dx &= \int\limits_{x^p}^{x^{p+1}} \phi_k \frac{\partial v}{\partial x}\, dx \\ &= \int\limits_{x^p}^{x^{p+1/2}} \phi_k \frac{\partial \phi_l}{\partial x}\hat{v}^p_l\, dx + \int\limits_{x^{p+1/2}}^{x^{p+1}} \phi_k \frac{\partial \phi_l}{\partial x}\hat{v}^{p+1}_l\, dx + \int\limits_{x^{p+1/2}}^{x^{p+1/2}} \delta\left(\phi_k \phi_l \hat{v}^{p+1/2}_l\right) d\ell,\end{aligned} \tag{19}$$

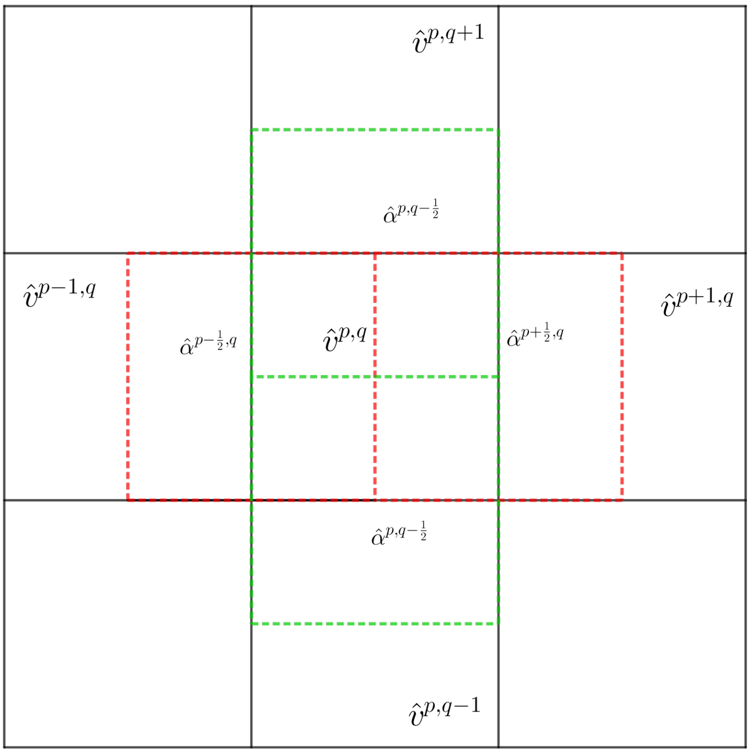
$\hat{v}^{p,q+1}$
$\hat{\alpha}^{p,q-\frac{1}{2}}$
$\hat{v}^{p-1,q}$
$\hat{\alpha}^{p-\frac{1}{2},q}$
$\hat{v}^{p,q}$
$\hat{\alpha}^{p+\frac{1}{2},q}$
$\hat{v}^{p+1,q}$
$\hat{\alpha}^{p,q-\frac{1}{2}}$
$\hat{v}^{p,q-1}$

Figure 1. Sketch of the discrete gradient and divergence operator.

where ϕ_k is a test function and $\delta(\cdot)$ is the Dirac function. The integral on the right hand side has been split into a left, a right and a jump contribution. Notice that for $\phi_k = \phi_l = 1$, the evaluation of (19) reduces to the jump term only, which corresponds to a second order central finite difference discretization of the derivative (18). The basis functions are always defined in the reference interval $\xi \in [0;1]$, however they must be shifted depending on the face-staggered element under consideration. Using the mapping (6) and performing a variable transformation in order to rescale all integrals to the reference unit element, the face-staggered gradient can be compactly written as

$$\mathbf{M}^x_{kl}\,\hat{\alpha}_l{}^{p+\frac{1}{2}} = \mathbf{R}^x_{kl}\,\hat{v}_l^{p+1} - \mathbf{L}^x_{kl}\,\hat{v}_l^{p}, \tag{20}$$

where the following matrices have been defined:

$$\mathbf{M}^x_{kl} = \int_0^1 \phi_k(\xi)\phi_l(\xi)\,d\xi, \tag{21a}$$

$$\mathbf{R}^x_{kl} = \phi_k\left(\frac{1}{2}\right)\phi_l(0) + \frac{1}{2}\int_0^1 \phi_k\left(\frac{\xi}{2}+\frac{1}{2}\right)\phi_l'\left(\frac{\xi}{2}\right)\,d\xi, \tag{21b}$$

$$\mathbf{L}^x_{kl} = \phi_k\left(\frac{1}{2}\right)\phi_l(1) - \frac{1}{2}\int_0^1 \phi_k\left(\frac{\xi}{2}\right)\phi_l'\left(\frac{\xi}{2}+\frac{1}{2}\right)\,d\xi, \tag{21c}$$

with $\phi_l' = \partial\phi_l/\partial\xi$. The extension to 3D can be done by considering the tensor product of matrices along each spatial direction. For instance, the 3D mass matrix can be assembled as follows:

$$\mathbf{M}^{xyz}_{kl} = \mathbf{M}^{\beta}_{m(\beta,k),m(\beta,l)}, \tag{22}$$

where $\beta = \{1,2,3\}$ indicates the spatial direction and $m(\beta,k), m(\beta,l)$ are multi-indexes that map the one-dimensional indexes $(\beta,k),(\beta,l)$ to the global index (k,l) in $\mathbf{M}^{xyz}_{kl}$. Notice that the one-dimensional mass matrix always corresponds to (21a) in each spatial direction, since it is defined in the reference interval. The 3D face-staggered gradient operator is then given along each direction by defining the following 3D matrices:

$$\mathbf{M}^{xyz}_{kl}\,\hat{\alpha}^{p+1/2,q,r}_{x,l} = \frac{\mathbf{M}^{yz}_{kl}}{\Delta x}\left(\mathbf{R}^{x}_{kl}\hat{v}^{p+1,q,r}_{l} - \mathbf{L}^{x}_{kl}\hat{v}^{p,q,r}_{l}\right) :=, \mathbf{R}^{3D,x}_{kl}\,\hat{v}^{p+1,q,r}_{l} - \mathbf{L}^{3D,x}_{kl}\,\hat{v}^{p,q,r}_{l} \tag{23a}$$

$$\mathbf{M}^{xyz}_{kl}\,\hat{\alpha}^{p,q+1/2,r}_{y,l} = \frac{\mathbf{M}^{xz}_{kl}}{\Delta y}\left(\mathbf{R}^{y}_{kl}\hat{v}^{p,q+1,r}_{l} - \mathbf{L}^{y}_{kl}\hat{v}^{p,q,r}_{l}\right) :=, \mathbf{R}^{3D,y}_{kl}\,\hat{v}^{p,q+1,r}_{l} - \mathbf{L}^{3D,y}_{kl}\,\hat{v}^{p,q,r}_{l}, \tag{23b}$$

$$\mathbf{M}^{xyz}_{kl}\,\hat{\alpha}^{p,q,r+1/2}_{z,l} = \frac{\mathbf{M}^{xy}_{kl}}{\Delta z}\left(\mathbf{R}^{z}_{kl}\hat{v}^{p,q,r+1}_{l} - \mathbf{L}^{z}_{kl}\hat{v}^{p,q,r}_{l}\right) :=, \mathbf{R}^{3D,z}_{kl}\,\hat{v}^{p,q,r+1}_{l} - \mathbf{L}^{3D,z}_{kl}\,\hat{v}^{p,q,r}_{l}. \tag{23c}$$

Now, the Laplace operator can be simply obtained as the divergence of the vector field $\boldsymbol{\alpha} = (\alpha_x, \alpha_y, \alpha_z)$, which indeed represents the gradient of the cell-centered scalar field $v(\boldsymbol{x},t)$ according to (18):

$$g = \nabla \cdot \boldsymbol{\alpha}. \tag{24}$$

Multiplication of the above equation by a test function, integration over the cell-centered control volume $C_{p,q,r}$ and application of Gauss theorem leads to

$$\begin{aligned}\int\limits_{C^{p,q,r}} \phi_k g \, dV &= \int\limits_{C^{p,q,r}} \phi_k \nabla \cdot \boldsymbol{\alpha} \, dV \\ &= \int\limits_{\partial C^{p,q,r}} \phi_k \boldsymbol{\alpha} \cdot \mathbf{n} \, dS - \int\limits_{C^{p,q,r}} \nabla \phi_k \, \boldsymbol{\alpha} \, dV,\end{aligned} \tag{25}$$

with $\partial C^{p,q,r}$ representing the boundary of the cell $C_{p,q,r}$ with outward pointing unit normal vector $\mathbf{n}$. On a Cartesian grid, one has $\mathbf{n}_x = (0,1,0)$, $\mathbf{n}_y = (0,0,1)$ and $\mathbf{n}_z = (0,0,1)$. Inserting the DG approximations $g = \phi_l \hat{g}_l$ and $\boldsymbol{\alpha} = \phi_l(\hat{\alpha}_{x,l}, \hat{\alpha}_{y,l}, \hat{\alpha}_{z,l})$ in the weak form (25), splitting the integral on the right hand side into all contributions belonging to each face-staggered control volume where $\boldsymbol{\alpha}$ is defined, and employing the explicit definitions of the outward normals, one gets

$$\begin{aligned}\mathbf{M}^{xyz}_{kl}\,\hat{g}^{p,q,r}_{l} = &\int\limits_{\partial C^{p+1/2,q,r}} \phi_k \phi_l \, \hat{\alpha}^{p+1/2,q,r}_{x,l} \, dS - \int\limits_{\partial C^{p-1/2,q,r}} \phi_k \phi_l \, \hat{\alpha}^{p-1/2,q,r}_{x,l} \, dS \\ + &\int\limits_{\partial C^{p,q+1/2,r}} \phi_k \phi_l \, \hat{\alpha}^{p,q+1/2,r}_{y,l} \, dS - \int\limits_{\partial C^{p,q-1/2,r}} \phi_k \phi_l \, \hat{\alpha}^{p,q-1/2,r}_{y,l} \, dS \\ + &\int\limits_{\partial C^{p,q,r+1/2}} \phi_k \phi_l \, \hat{\alpha}^{p,q,r+1/2}_{z,l} \, dS - \int\limits_{\partial C^{p,q,r-1/2}} \phi_k \phi_l \, \hat{\alpha}^{p,q,r-1/2}_{z,l} \, dS \\ - &\left(\int\limits_{C^{p,q,r}} \frac{\partial \phi_k}{\partial x}\alpha_x + \frac{\partial \phi_k}{\partial y}\alpha_y + \frac{\partial \phi_k}{\partial z}\alpha_z \, dV \right).\end{aligned} \tag{26}$$

Notice that for $\phi_k = \phi_l = 1$, the divergence operator $\mathbb{D}^{p,q,r}$ given by (14) is retrieved with second order accuracy, since the volume integrals vanish. Because the vector field $\boldsymbol{\alpha}$ is defined on the face-staggered mesh, the volume integrals appearing in (26) must be split into a left and a right contribution. For instance, in the $x-$direction we have two sub-volumes given by $C^{p,q,r}_{x,L} = [x^{p-1/2}; x^{p}] \times [y^{q-1/2}; y^{q+1/2}] \times [z^{r-1/2}; z^{r+1/2}]$ and $C^{p,q,r}_{x,R} = [x^{p}; x^{p+1/2}] \times [y^{q-1/2}; y^{q+1/2}] \times [z^{r-1/2}; z^{r+1/2}]$, that is

$$\int\limits_{C^{p,q,r}} \frac{\partial \phi_k}{\partial x}\alpha_x \, dV = \int\limits_{C^{p,q,r}_{x,L}} \frac{\partial \phi_k}{\partial x}\phi_l \hat{\alpha}^{p-1/2,q,r}_{x,l} \, dV - \int\limits_{C^{p,q,r}_{x,R}} \frac{\partial \phi_k}{\partial x}\phi_l \hat{\alpha}^{p+1/2,q,r}_{x,l} \, dV. \tag{27}$$

Applying the same strategy to all spatial directions, mapping the integrals to the reference system with consequently shifting the basis functions according to the face-staggering where the jump integrals of (26) are defined, and using the 3D matrix notation introduced in (23), the Laplace operator $\mathbb{L}^{p,q,r}$ in the semi-implicit momentum equation (10a) can be eventually discretized as follows:

$$\begin{aligned}\mathbb{L}^{p,q,r}(g) &= \mathbf{M}^{xyz}_{kl}\,\hat{g}^{p,q,r}_l \\ &= \left[\left(\mathbf{L}^{3D,x}_{kl}\right)^\top \hat{\alpha}^{p+1/2,q,r}_{x,l} - \left(\mathbf{R}^{3D,x}_{kl}\right)^\top \hat{\alpha}^{p-1/2,q,r}_{x,l}\right]\frac{1}{\Delta x} \\ &+ \left[\left(\mathbf{L}^{3D,y}_{kl}\right)^\top \hat{\alpha}^{p,q+1/2,r}_{y,l} - \left(\mathbf{R}^{3D,y}_{kl}\right)^\top \hat{\alpha}^{p,q-1/2,r}_{y,l}\right]\frac{1}{\Delta y} \\ &+ \left[\left(\mathbf{L}^{3D,z}_{kl}\right)^\top \hat{\alpha}^{p,q,r+1/2}_{z,l} - \left(\mathbf{R}^{3D,z}_{kl}\right)^\top \hat{\alpha}^{p,q,r-1/2}_{z,l}\right]\frac{1}{\Delta z}.\end{aligned} \tag{28}$$

The Laplace operator is symmetric and positive definite [39], therefore the implicit viscous sub-system (15) can be solved using the conjugate gradient method in its matrix-free implementation.

3.3. *High order extension in time*

Time discretization is enforced with high order of accuracy relying on the class of semi-implicit IMEX schemes originally developed in [43], which have been proven to be a powerful tool in the context of compressible gas dynamics [37]. First, the governing equations (1) are written under the form of an autonomous system, that is

$$\frac{\partial \boldsymbol{u}}{\partial t} = \mathcal{H}\left(\boldsymbol{u}(t), \boldsymbol{u}(t)\right), \qquad \forall t > t_0, \qquad \text{with} \qquad \boldsymbol{u}(t_0) = \boldsymbol{u}_0, \tag{29}$$

where $\boldsymbol{u}_0$ defines the initial condition at time t_0. The function $\mathcal{H}$ represents the spatial approximation of the terms $\nabla \cdot \mathbf{F}_c + \nabla p - \nu\nabla^2 \boldsymbol{u}$ in (1), which is given by the operators presented in the previous sections. The first argument of $\mathcal{H}$ is discretized explicitly and it is labeled with $\boldsymbol{u}_E$, while the second argument $\boldsymbol{u}_I$ is taken implicitly as well as the pressure p_I. A partitioned system is then retrieved with $\boldsymbol{u} = (\boldsymbol{u}_E, \boldsymbol{u}_I)$, hence

$$\begin{cases} \dfrac{\partial \boldsymbol{u}_E}{\partial t} = \mathcal{H}\left(\boldsymbol{u}_E, \boldsymbol{u}_I\right) \\ \dfrac{\partial \boldsymbol{u}_I}{\partial t} = \mathcal{H}\left(\boldsymbol{u}_E, \boldsymbol{u}_I\right) \end{cases}. \tag{30}$$

As investigated in [43], the number of unknowns has not been doubled because the governing PDE can be written in the form of an autonomous system and we will choose stiffly accurate time discretizations. The incompressible Navier-Stokes equations can therefore be cast into the formalism (30) by defining

$$\mathcal{H}\left(\boldsymbol{u}_E, \boldsymbol{u}_I\right) = \begin{cases} -(\boldsymbol{u}\otimes\boldsymbol{u})_E - p_I + \nu\nabla\boldsymbol{u}_I \\ \boldsymbol{u}_I \end{cases}. \tag{31}$$

High order in time is achieved making use of implicit-explicit (IMEX) Runge-Kutta schemes [29], that are multi-step methods based on a total number of s stages that depend on the accuracy and stability property of the chosen IMEX-RK scheme. The time marching takes then the following structure. First we set $\boldsymbol{u}^n_E = \boldsymbol{u}^n_I = \boldsymbol{u}^n$, then the stage fluxes for $i = 1, \ldots, s$ are computed in the following way:

$$\boldsymbol{u}^i_E = \boldsymbol{u}^n_E + \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} k_j, \qquad 2 \le i \le s, \tag{32a}$$

$$\tilde{\boldsymbol{u}}^i_I = \boldsymbol{u}^n_E + \Delta t \sum_{j=1}^{i-1} a_{ij} k_j, \qquad 2 \le i \le s, \tag{32b}$$

$$k_i = \mathcal{H}\left(\boldsymbol{u}^i_E, \tilde{\boldsymbol{u}}^i_I + \Delta t\, a_{ii}\, k_i\right), \qquad 1 \le i \le s. \tag{32c}$$

The coefficients of the explicit and implicit Runge-Kutta method are normally described with the double Butcher tableau:

$$\begin{array}{c|c} \tilde{c} & \tilde{A} \\ \hline & \tilde{b}^\top \end{array} \qquad \begin{array}{c|c} c & A \\ \hline & b^\top \end{array}, \tag{33}$$

with the matrices $(\tilde{A}, A) \in \mathbb{R}^{s\times s}$ and the vectors $(\tilde{c}, c, \tilde{b}, b) \in \mathbb{R}^s$. The tilde symbol refers to the explicit scheme and matrix $\tilde{A} = (\tilde{a}_{ij})$ is a lower triangular matrix with zero elements on the diagonal, while $A = (a_{ij})$ is a triangular matrix which accounts for the implicit scheme, thus having non-zero elements on the diagonal. The implicit part to the RK scheme is in fact linearly implicit, since the A matrix is triangular. This clearly makes the resolution simpler than a fully implicit RK time stepping technique. Applying the partitioned Runge-Kutta method to (31) under the assumption that the system is autonomous, only one set of stage fluxes needs to be computed by solving the linear system (32c), that is nothing but the elliptic equation on the pressure (17) obtained after substitution of the momentum equation into the divergence-free constraint. We adopt stiffly accurate schemes [43], thus the new solution $\boldsymbol{u}_E^{n+1} = \boldsymbol{u}_I^{n+1} = \boldsymbol{u}^{n+1}$ is simply given by the last stage of the Runge-Kutta algorithm (32), i.e. $\boldsymbol{u}^{n+1} = \boldsymbol{u}^s$. If the IMEX method is chosen to be not stiffly accurate, then the solution at the new time level can be computed as

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \Delta t \sum_{i=1}^{s} b_i \, k_i. \tag{34}$$

3.4. *Implementation and optimization*

The SI-IMEX numerical method is implemented in a code written with `Fortran` language. The parameters and settings of the ongoing computation (viscosity coefficient, CFL number, mesh discretization, ...) can then be conveniently fixed at the aid of pre-processors suitably distributed throughout the code, hence compiling only the part of code referred to the chosen setup.

The total computational time of those simulations can grow exponentially due to a high mesh granularity or an increasing in the degree of the numerical approximation, for instance. That is the reason why we start optimizing the code by exploiting highly optimized libraries of Math Kernel Library (MKL) of Intel. Specifically, since both high order reconstruction and Laplace operators are carried out in the reference system, all integrals of the basis functions can be pre-computed and stored once and for all before the beginning of the time evolution loop. As a consequence, during the computation we mainly have to deal with matrix-matrix and matrix-vector multiplications.

In order to increase the performance, we used Intel®VTune™Profiler, which is an Intel API that can optimize performance of applications running on Intel processors. This tool shows the execution time of every subroutine and every instruction executed by the code in a completely isolated environment, and by doing so we found the most expensive lines of the code. As expected, those were operations involving matrix-matrix and matrix-vector products computed with `MATMUL` intrinsic function. Next, we replaced these instructions by `DGEMM` subroutines of the Intel MKL library widely employed in HPC system. This simple substitution allowed us to remarkably improve the overall algorithm efficiency, halving the execution time of the most expensive subroutines and almost eliminating the cost of other minor functions.

Then we noticed very long computational time in the access to some variables like rank-4 arrays, in particular with the array containing the Voronoi neighborhood of every grid element. In order to reduce the access time, we deconstructed this array adding a new rank-2 array which reordered all neighbors in one single row for each grid cell. In this way, when the code has to read inside the neighbor array, it can use a rank-1 variable instead of a rank-3 making the access almost instantaneous.

Finally, we took care about other minor optimizations: we replaced some `DO`-loops with instructions that take advantage from the processor vectorization and we introduced new temporary variables with the aim of speeding up the access to other frequently used arrays. Those operations allow the code to significantly reduce its total computational running time.

To distribute memory consumption and improve the overall efficiency of the computation, the code is parallelized with MPI library in order to exploit multi-processor architectures.

4. Numerical results

In this section we present some academic benchmarks in order to verify the accuracy and the robustness of the novel semi-implicit IMEX schemes (SI-IMEX) for the solution of the incompressible Navier-Stokes model (1). The time step is computed according to (3) with CFL = 0.9, and all computations are carried out in a fully 3D domain using MPI parallelization on 64 Intel SkyLake CPUs.

4.1. *2D Taylor-Green vortex*

The convergence rates of the novel SI-IMEX schemes are studied by considering the 2D Taylor-Green vortex test, which has an exact solution in the incompressible Navier-Stokes limit:

$$\begin{aligned} u(x,y,t) &= \sin(x)\cos(y)\,e^{-2\nu t}, \\ v(x,y,t) &= \cos(x)\sin(y)\,e^{-2\nu t}, \\ w(x,y,t) &= 0, \\ p(x,y,t) &= 0.25\,(\cos(2x)+\cos(2y))\;e^{-4\nu t}. \end{aligned} \tag{35}$$

The computational domain is given by $\Omega = [0;2\pi]^2 \times [0;1]$ with periodic boundaries imposed everywhere, and it is discretized with $N_z = 4$ and a sequence of successively refined meshes on the $x-y$ plane. The simulation is run until the final time $t_f = 0.2$ with two different values of viscosity, namely $\nu = 10^{-5}$ and $\nu = 10^{-2}$. The errors are measured for the velocity components u, v and pressure p in L_2 norms, as reported in Tables 1-2. The formal accuracy is very well retrieved for second and third order schemes independently of the chosen viscosity coefficient.

Table 1. Numerical convergence results of the SI-IMEX scheme with second and third order of accuracy in space and time using the 2D Taylor-Green vortex problem with $\nu = 10^{-5}$. The errors are measured in L_1, L_2 and L_∞ norms and refer to the velocity component u at time $t = 0.2$.

SI-IMEX $\mathcal{O}(2)$, $\nu = 10^{-5}$

$N_x = N_y$	u_{L_2}	$\mathcal{O}(u_{L_2})$	v_{L_2}	$\mathcal{O}(v_{L_2})$	p_{L_2}	$\mathcal{O}(p_{L_2})$
25	3.288E-02	-	3.288E-02	-	7.167E-02	-
50	8.259E-03	1.99	8.259E-03	1.99	1.625E-02	2.14
100	2.067E-03	2.00	2.067E-03	2.00	4.098E-03	1.99
200	5.167E-04	2.00	5.167E-04	2.00	1.009E-03	2.02

SI-IMEX $\mathcal{O}(3)$, $\nu = 10^{-5}$

$N_x = N_y$	u_{L_2}	$\mathcal{O}(u_{L_2})$	v_{L_2}	$\mathcal{O}(v_{L_2})$	p_{L_2}	$\mathcal{O}(p_{L_2})$
25	3.478E-03	-	3.478E-03	-	1.082E-02	-
50	4.246E-04	3.03	4.246E-04	3.03	1.212E-03	2.99
100	5.302E-05	3.00	5.302E-05	3.00	1.525E-04	2.99
200	6.628E-06	3.00	6.628E-06	3.00	2.887E-05	2.73

Figure 2 depicts the third order numerical solution obtained at the final time on a computational mesh with $N_x = N_y = 200$, showing no spurious oscillation and almost retrieving the analytical solution.

Finally, in Table 3 we show the computational times for all simulations carried out for the convergence study. The optimized code with MKL is compared against the non-optimized version, demonstrating the effectiveness of the improvements related to the implementation of the SI-IMEX schemes. For third order and high viscosity ($\nu = 10^{-2}$), the optimized code achieves a gaining factor greater than 4, while for a lower viscosity ($\nu = 10^{-5}$) the optimized code runs approximately in half time the same simulations with respect to the non-optimized version. A visual representation of the obtained results is conveniently shown in Figure 3.

We also measure the performance of our code on the HPC architecture *Galileo100* (Cineca - Italy). We run the 2D Taylor-Green test on a mesh composed of 2750×2750×4 cells with $N = 2$, hence involving

Table 2. Numerical convergence results of the SI-IMEX scheme with second and third order of accuracy in space and time using the 2D Taylor-Green vortex problem with $\nu = 10^{-2}$. The errors are measured in L_1, L_2 and L_∞ norms and refer to the velocity component u at time $t = 0.2$.

SI-IMEX $O(2)$, $\nu = 10^{-2}$

$N_x = N_y$	u_{L_2}	$O(u_{L_2})$	v_{L_2}	$O(v_{L_2})$	p_{L_2}	$O(p_{L_2})$
25	3.282E-02	-	3.282E-02	-	6.275E-02	-
50	8.243E-03	1.99	8.243E-03	1.99	1.239E-02	2.34
100	2.062E-03	2.00	2.062E-03	2.00	2.750E-03	2.17
200	5.157E-04	2.00	5.157E-04	2.00	9.204E-04	1.58

SI-IMEX $O(3)$, $\nu = 10^{-2}$

$N_x = N_y$	u_{L_2}	$O(u_{L_2})$	v_{L_2}	$O(v_{L_2})$	p_{L_2}	$O(p_{L_2})$
25	3.489E-03	-	3.489E-03	-	1.067E-02	-
50	4.263E-04	3.03	4.263E-04	3.03	1.209E-03	2.98
100	5.339E-05	3.00	5.339E-05	3.00	1.527E-04	2.99
200	6.699E-06	2.99	6.699E-06	2.99	3.313E-05	2.61

Table 3. Computational time measured in seconds with and without code optimization with MKL for the 2D Taylor-Green vortex problem with $\nu = \{10^{-2}, 10^{-5}\}$ with second and third order of accuracy SI-IMEX schemes. The ratio β between the computational times is also reported.

SI-IMEX $O(2)$

	$\nu = 10^{-2}$			$\nu = 10^{-5}$		
$N_x = N_y$	no MKL	MKL	β	no MKL	MKL	β
25	6.844E+00	3.766E+00	1.82	6.594E+00	3.641E+01	1.80
50	5.372E+01	2.997E+01	1.79	5.178E+01	3.080E+02	1.78
100	4.299E+02	2.450E+02	1.75	4.182E+02	2.446E+03	1.74
200	3.462E+03	1.983E+03	1.75	3.347E+03	1.822E+04	1.74

SI-IMEX $O(3)$

	$\nu = 10^{-2}$			$\nu = 10^{-5}$		
$N_x = N_y$	no MKL	MKL	β	no MKL	MKL	β
25	6.844E+01	1.623E+01	4.22	3.641E+01	1.233E+01	2.95
50	5.846E+02	1.355E+02	4.31	3.080E+02	9.939E+01	3.10
100	4.930E+03	1.121E+03	4.40	2.446E+03	7.921E+02	3.09
200	4.093E+04	9.212E+03	4.44	1.822E+04	6.161E+03	2.96

a total number of 816′750′000 degrees of freedom. We perform a strong scaling using from 1 to 192 CPUs. Table 4 reports the computational times of each simulation as well as the percentage related to MPI communications. Speedup, efficiency and Kuck functions are also presented, and Figure 4 highlights that the code could achieve an efficiency of approximately 0.75 on 192 cores.

Table 4. Data collected and processed on the *Galileo100* architecture (Cineca - Italy) for strong scaling test of the SI-IMEX code with 816′750′000 degrees of freedom.

NCPU	Time (s)	% MPI	Speedup	Efficiency	Kuck function
1	2.063E+05	-	1.000E+00	1.000E+00	1.000E+00
24	9.477E+03	9.29%	2.178E+01	9.075E-01	1.977E+01
48	4.849E+03	14.07%	4.168E+01	8.685E-01	3.620E+01
96	2.551E+03	18.51%	7.753E+01	8.076E-01	6.261E+01
192	1.411E+03	25.67%	1.432E+02	7.305E-01	1.046E+02

4.2. *Double shear layer*

Another classical problem for incompressible solvers is the double shear layer as proposed in [46]. The problem consists in an horizontal jet that is initially perturbed in a periodic domain. In order to reduce to

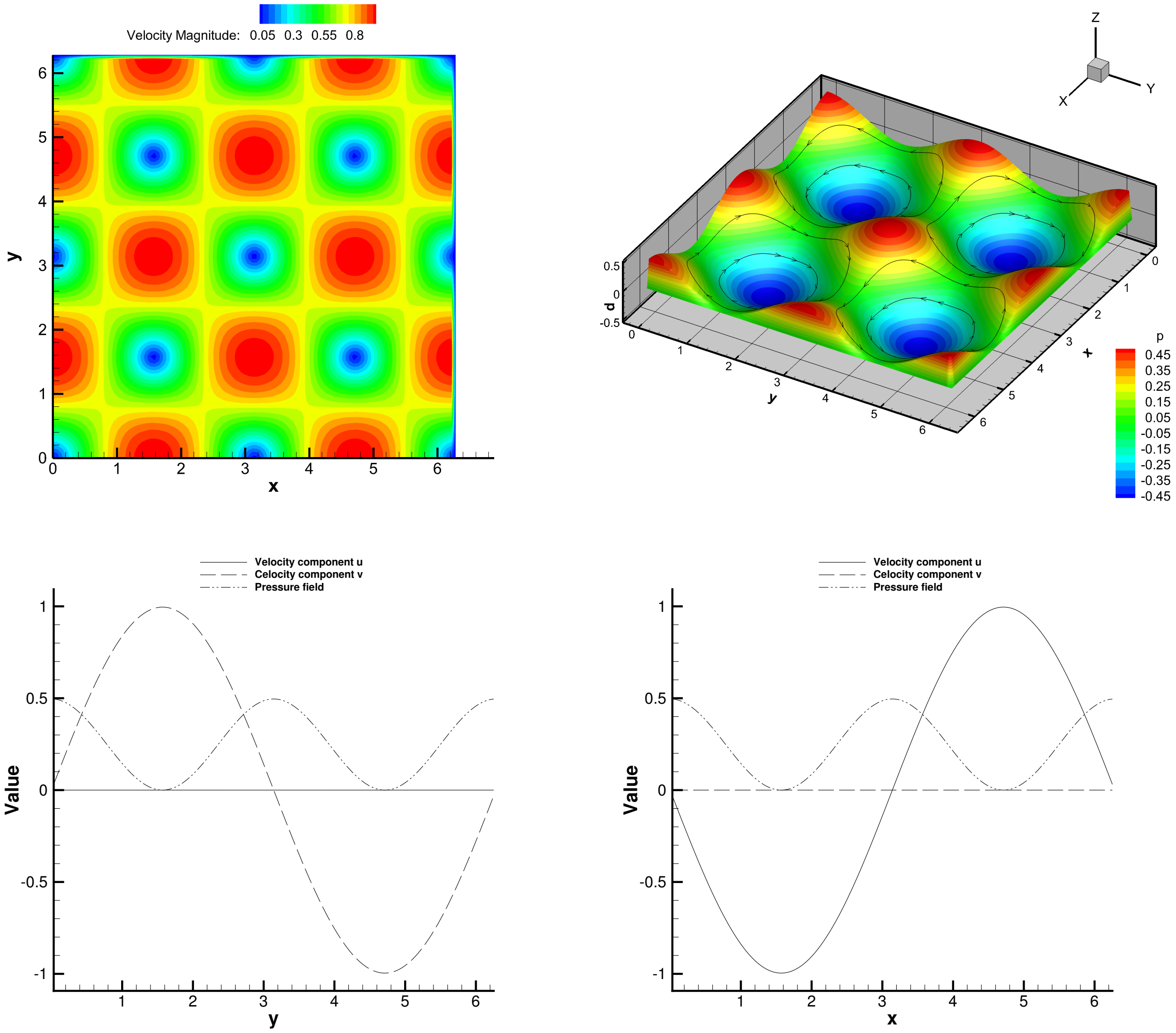

Figure 2. Numerical results for the 2D Taylor-Green vortex problem at $t_f = 0.2$ with $\nu = 10^{-2}$. Velocity magnitude (top left), pressure field and stream-traces (top right), main quantities computed along the lines $x = \pi$ (bottom left) and $y = \pi$ (bottom right).

the two-dimensional case, the computational domain is $\Omega = [0,1]\times[0,0.1]\times[0,1]$ with periodic boundary conditions everywhere, and it is covered with a Cartesian mesh composed of $N_e = 160\times 4\times 160$ control volumes, hence using a coarser resolution compared to [32,47]. The velocity field is initialized with

$$u = \begin{cases} \tanh\left(\theta\,(z-0.25)\right) & \text{if } z \leq 0.5 \\ \tanh\left(\theta\,(0.75-z)\right) & \text{if } z > 0.5 \end{cases}, \qquad v = 0, \qquad w = \delta\sin(2\pi x), \tag{36}$$

with $\theta = 30$ and $\delta = 0.05$. The viscosity coefficient is set to $\nu = 5\cdot 10^{-3}$, while the final time of the simulation is $t_f = 1.8$. The evolution of the vorticity magnitude is reported in Figure 5 showing the correct reproduction of the vortical patterns, which are in line with those ones available in the literature [47,48]

4.3. *Lid-driven cavity*

We now want to test our algorithm against another classical benchmark for incompressible solvers, namely the lid-driven cavity test case. We consider a domain $\Omega = [-0.5,0.5]\times[-0.5,0.5]\times$

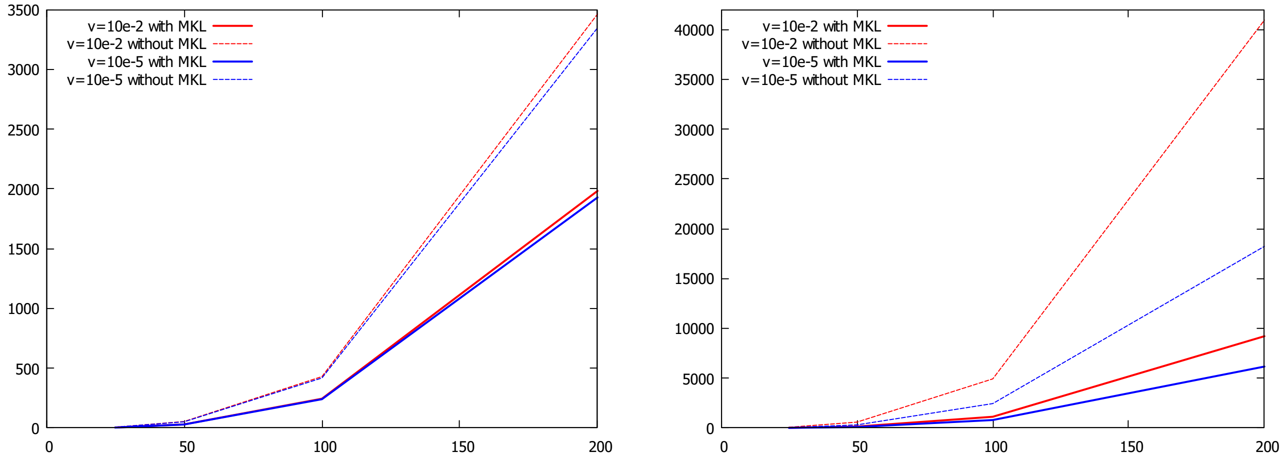

Figure 3. Comparison between the computational times measured in seconds in running the 2D Taylor-Green vortex test with and without MKL optimization using second order ($\mathcal{O}(2)$, on the left) and third order of accuracy SI-IMEX schemes ($\mathcal{O}(3)$, on the right).

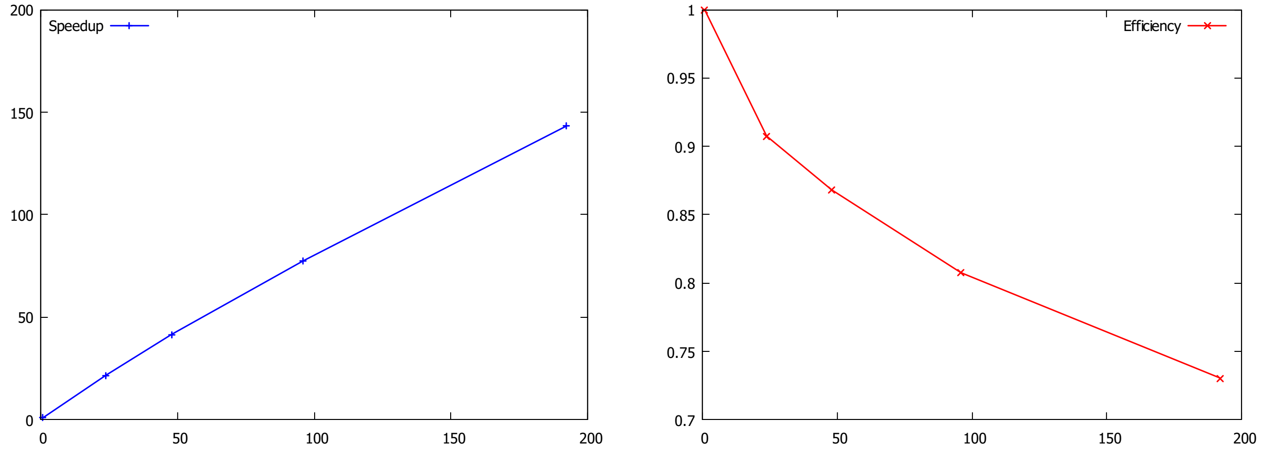

Figure 4. Strong scaling test with MPI of the SI-IMEX schemes. Speedup (left) and efficiency (right) obtained running the 2D Taylor-Green vortex test on a grid with $2750 \times 2750 \times 4$ elements, using third order of space accuracy. The total number of degrees of freedom is $816'750'000$.

$[-0.00625, 0.00625]$ in order to approach a two-dimensional cavity, covered with $N_e = 160 \times 160 \times 4$ elements. Since we are looking for a steady state solution, we use a third order space discretization, i.e. $N = 2$, while adopting a cheaper first order in time scheme. All the quantities are set initially to zero. An horizontal velocity $\boldsymbol{u} = (1, 0, 0)$ is imposed on the upper face of the cavity ($y = 0.5$), and it is the only responsible of the motion in the cavity induced by convection and viscous forces. No-slip wall boundary conditions are imposed on the remaining sides in the x and y directions, while periodic boundary conditions are assumed in the $z-$direction. To test the novel implicit solver for the viscous sub-system, we set a rather high value for the viscosity $\nu = 10^{-2}$ that corresponds to a Reynolds number of $Re = 100$. The simulation runs up to $t_f = 25$. The resulting velocity distribution is reported in Figure 6 together with the one-dimensional plot of the perpendicular velocities along the two lines located at $(x, 0, 0)$ and $(0, y, 0)$. In this case, the numerical solution computed with the SI-IMEX scheme is compared with the reference solution that was obtained by Ghia et.al. in [49], achieving a quite good matching.

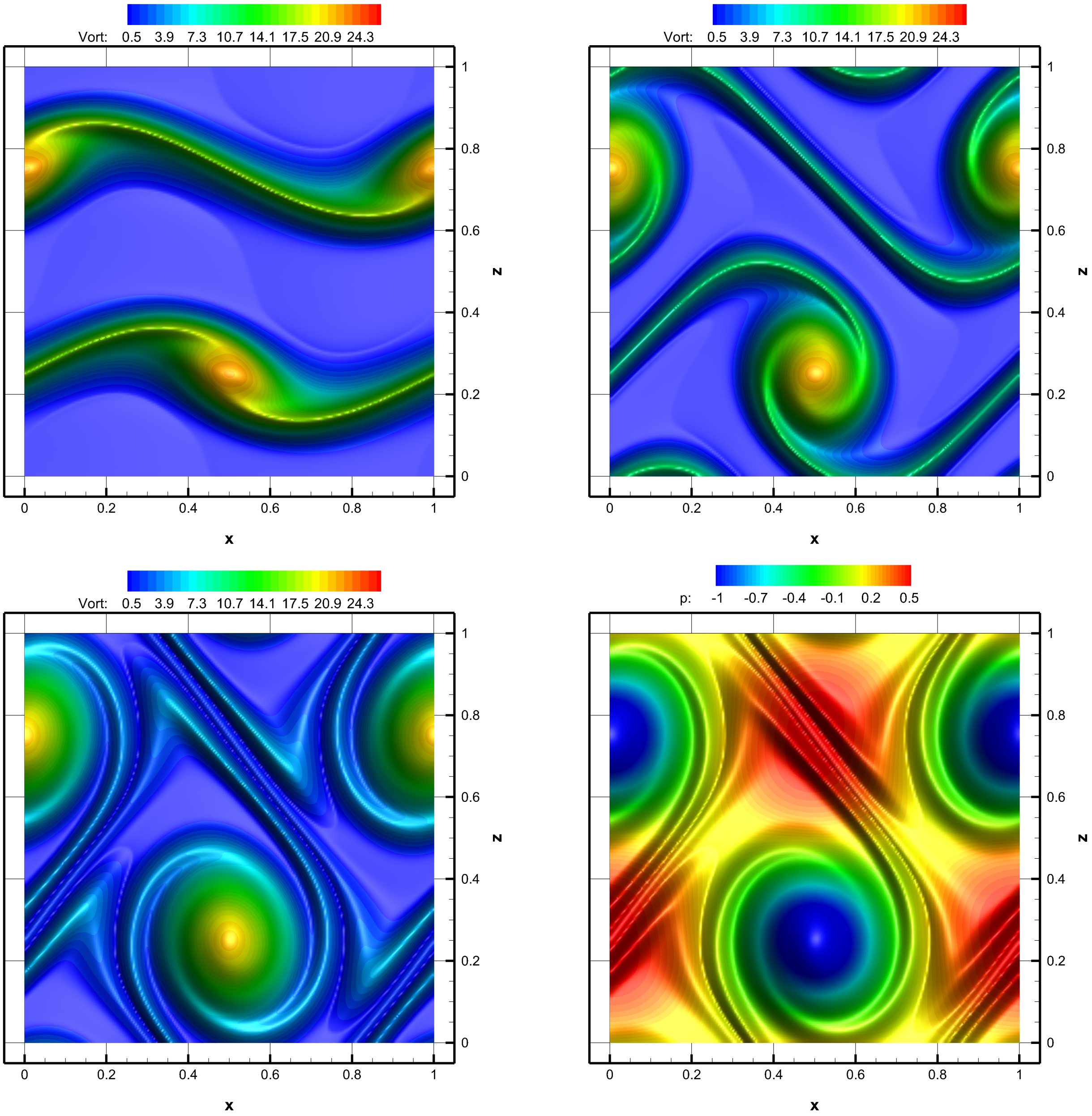

Figure 5. Numerical results for the double shear layer test. Vorticity magnitude at output times $t = 0.8$ (top left), $t = 1.2$ (top right), $t = 1.8$ (bottom left) and pressure field at the final time $t = 1.8$ (bottom right).

5. Conclusions

In this article we have presented a semi-implicit IMEX scheme for the solution of the incompressible Navier-Stokes equations on fully three-dimensional computational domains with Cartesian grids. High order of accuracy in space is achieved using a finite difference scheme in flux form combined with an efficient CWENO reconstruction which is carried out in a dimension-by-dimension fashion. The class of semi-implicit IMEX schemes is adopted to increase the accuracy of the method in time as well. Convective terms are discretized explicitly, while the pressure gradient is solved implicitly. The momentum equation is directly inserted into the divergence-free constraint of the velocity field, hence obtaining an elliptic equation on the pressure which satisfies up to machine precision the involution of the PDE system. An implicit treatment of the the viscous terms is proposed, which is based on a weak formulation of the gradient and divergence operators related to the parabolic viscous contribution in the momentum

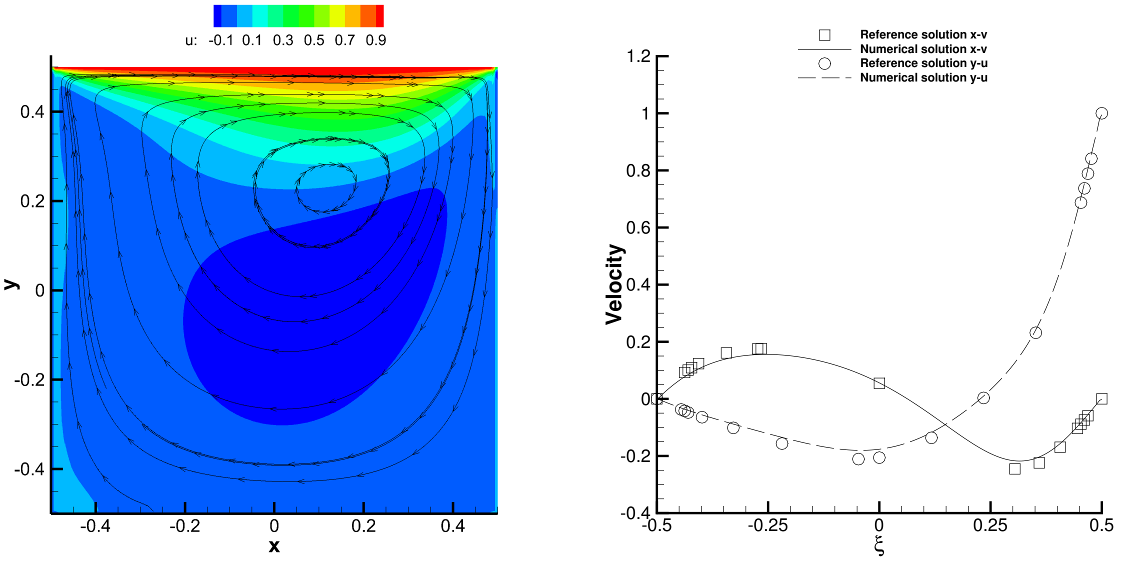

Figure 6. Obtained velocity component u (left) and comparison with the reference solution of Ghia [49] on the central lines $(x, 0, 0)$ and $(0, y, 0)$ (right).

equation. A local discontinuous Galerkin scheme is then derived to obtain a symmetric and positive definite Laplace operator, that is exploited to resolve the viscous effects implicitly. Eventually, the time step is restricted by a CFL-type stability condition which is only based on the fluid velocity and not on the eigenvalues of the viscous sub-system of the governing PDE. Particular care has also been devoted to the optimization of the code relying on highly optimized Intel MKL routines and MPI parallelization.

Future research will concern the extension of the proposed approach to the Saint-Venant equations in the context of environmental flows [23,50], with the usage of unstructured orthogonal meshes [51]. Possible applications to the volume of fluid method [52] are also foreseen.

Acknowledgments

WB would like to thank the Italian Ministry of Instruction, University and Research (MIUR) to support this research with funds coming from PRIN Project 2017 (No. 2017KKJP4X entitled Â“Innovative numerical methods for evolutionary partial differential equations and applicationsÂ”). W.B. and M.T. are members of the GNCS-INdAM (Italian Institute of high Mathematics) group. The authors acknowledge the Cineca supercomputing center for granting access to the architecture *Galileo100* under the project IsC96 (2022).

References

1. E. Ferrer and R. Willden, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *International Journal of Heat and Mass Transfer*, vol. 15, pp. 1787–1806, 1972.
2. C. Taylor and P. Hood, A numerical solution of the Navier-Stokes equations using the finite element technique, *Computers and Fluids*, vol. 1, pp. 73–100, 1973.
3. A. Brooks and T. Hughes, Stream-line upwind/Petrov Galerkin formulstion for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equation, *Computer Methods in Applied Mechanics and Engineering*, vol. 32, pp. 199–259, 1982.
4. T. Hughes, M. Mallet, and M. Mizukami, A new finite element formulation for computational fluid dynamics: II. Beyond SUPG, *Computer Methods in Applied Mechanics and Engineering*, vol. 54, pp. 341–355, 1986.

5. M. Fortin, Old and new finite elements for incompressible flows, *International Journal for Numerical Methods in Fluids*, vol. 1, pp. 347–364, 1981.
6. R. Verfürth, Finite element approximation of incompressible Navier-Stokes equations with slip boundary condition II, *Numerische Mathematik*, vol. 59, pp. 615–636, 1991.
7. J. G. Heywood and R. Rannacher, Finite element approximation of the nonstationary Navier-Stokes Problem. I. Regularity of solutions and second order error estimates for spatial discretization, *SIAM Journal on Numerical Analysis*, vol. 19, pp. 275–311, 1982.
8. J. G. Heywood and R. Rannacher, Finite element approximation of the nonstationary Navier-Stokes Problem. III. Smoothing property and higher order error estimates for spatial discretization, *SIAM Journal on Numerical Analysis*, vol. 25, pp. 489–512, 1988.
9. F. Harlow and J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface, *Physics of Fluids*, vol. 8, pp. 2182–2189, 1965.
10. V. Patankar and B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *International Journal of Heat and Mass Transfer*, vol. 15, pp. 1787–1806, 1972.
11. S. Patankar, *Numerical heat transfer and fluid flow.* New York: McGraw-Hill, 1980.
12. J. van Kan, A second-order accurate pressure correction method for viscous incompressible flow, *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 870–891, 1986.
13. F. Bassi, A. Crivellini, D. D. Pietro, and S. Rebay, An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows, *Computers and Fluids*, vol. 36, pp. 1529–1546, 2007.
14. K. Shahbazi, P. F. Fischer, and C. R. Ethier, A high-order discontinuous galerkin method for the unsteady incompressible navier-stokes equations, *Journal of Computational Physics*, vol. 222, pp. 391–407, 2007.
15. E. Ferrer and R. Willden, A high order Discontinuous Galerkin Finite Element solver for the incompressible Navier-Stokes equations, *Computer and Fluids*, vol. 46, pp. 224–230, 2011.
16. N. Nguyen, J. Peraire, and B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations, *Journal of Computational Physics*, vol. 230, pp. 1147–1170, 2011.
17. S. Rhebergen and B. Cockburn, A space-time hybridizable discontinuous Galerkin method for incompressible flows on deforming domains, *Journal of Computational Physics*, vol. 231, pp. 4185–4204, 2012.
18. S. Rhebergen, B. Cockburn, and J. J. van der Vegt, A space-time discontinuous Galerkin method for the incompressible Navier-Stokes equations, *Journal of Computational Physics*, vol. 233, pp. 339–358, 2013.
19. A. Crivellini, V. D'Alessandro, and F. Bassi, High-order discontinuous Galerkin solutions of three-dimensional incompressible RANS equations, *Computers and Fluids*, vol. 81, pp. 122–133, 2013.
20. B. Klein, F. Kummer, and M. Oberlack, A SIMPLE based discontinuous Galerkin solver for steady incompressible flows, *Journal of Computational Physics*, vol. 237, pp. 235–250, 2013.
21. A. Chorin, A numerical method for solving incompressible viscous flow problems, *Journal of Computational Physics*, vol. 2, pp. 12–26, 1967.
22. F. Bassi, A. Crivellini, D. D. Pietro, and S. Rebay, On a robust discontinuous Galerkin technique for the solution of compressible flow, *Journal of Computational Physics*, vol. 218, pp. 208–221, 2006.
23. V. Casulli, Semi-implicit finite difference methods for the two–dimensional shallow water equations, *J. Comp. Phys.*, vol. 86, pp. 56–74, 1990.
24. V. Casulli and R. Cheng, Semi-implicit finite difference methods for three-dimensional shallow water flow, *Int. J. Numer. Methods Fluids*, vol. 15, pp. 629–648, 1992.
25. M. Tavelli and M. Dumbser, A staggered space-time discontinuous Galerkin method for the three-dimensional incompressible Navier-Stokes equations on unstructured tetrahedral meshes, *J. Comp. Phys.*, vol. 319, pp. 294–323, 2016.

26. U. M. Ascher, S. J. Ruuth, and R. J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, *Appl. Numer. Math.*, vol. 25, pp. 151–167, 1982.

27. S. Boscarino and L. Pareschi, On the asymptotic properties of IMEX Runge-Kutta schemes for hyperbolic balance laws, *Journal of Computational and Applied Mathematics*, vol. 316, pp. 60–73, 2017.

28. S. Boscarino and G. Russo, On a class of uniformly accurate IMEX Runge-Kutta schemes and applications to hyperbolic systems with relaxation, *SIAM J. Sci. Comput.*, vol. 31, pp. 1926–1945, 2009.

29. L. Pareschi and G. Russo, Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation, *J. Sci. Comput.*, vol. 25, pp. 129–155, 2005.

30. S. Boscarino, L. Pareschi, and G. Russo, A unified IMEX Runge-Kutta approach for hyperbolic systems with multiscale relaxation, *SIAM J. Numer. Anal.*, vol. 55, no. 4, pp. 2085–2109, 2017.

31. W. Boscheri, G. Dimarco, R. Loubère, M. Tavelli, and M. Vignal, A second order all Mach number IMEX finite volume solver for the three dimensional Euler equations, *J. Comp. Phys.*, vol. 415, p. 109486, 2020.

32. W. Boscheri, G. Dimarco, and M. Tavelli, An efficient second order all Mach finite volume solver for the compressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering*, vol. 374, p. 113602, 2021.

33. V. DeCaria and M. Schneier, An embedded variable step IMEX scheme for the incompressible Navier-Stokes equations, *Computer Methods in Applied Mechanics and Engineering*, vol. 376, p. 113661, 2021.

34. F. Meng, J. Banks, W. Henshaw, and D. Schwendeman, Fourth-order accurate fractional-step IMEX schemes for the incompressible Navier-Stokes equations on moving overlapping grids, *Computer Methods in Applied Mechanics and Engineering*, vol. 366, p. 113040, 2020.

35. W. Wang, Z. Wang, and M. Mao, Linearly implicit variable step-size BDF schemes with Fourier pseudospectral approximation for incompressible Navier-Stokes equations, *Applied Numerical Mathematics*, vol. 172, pp. 393–412, 2022.

36. A. Larios, L. G. Rebholz, and C. Zerfas, Global in time stability and accuracy of imex-fem data assimilation schemes for navier-stokes equations, *Computer Methods in Applied Mechanics and Engineering*, vol. 345, pp. 1077–1093, 2019.

37. W. Boscheri and L. Pareschi, High order pressure-based semi-implicit IMEX schemes for the 3D Navier-Stokes equations at all Mach numbers, *J. Comp. Phys.*, vol. 434, p. 110206, 2021.

38. D. Levy, G. Puppo, and G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, *M2AN Math. Model. Numer. Anal.*, vol. 33, no. 3, pp. 547–571, 1999.

39. F. Fambri and M. Dumbser, Spectral semi-implicit and space-time discontinuous Galerkin methods for the incompressible Navier-Stokes equations on staggered Cartesian grids, *Applied Numerical Mathematics*, vol. 110, pp. 41–74, 2016.

40. A. Stroud, *Approximate Calculation of Multiple Integrals.* Englewood Cliffs, New Jersey: Prentice-Hall Inc., 1971.

41. V. Casulli, A semi-implicit finite difference method for non-hydrostatic free-surface flows, *Int. J. Num. Meth. in Fluids*, vol. 30, pp. 425–440, 1999.

42. B. Einfeldt, C. Munz, P. Roe, and B. Sjögreen, A pressure-based semi-implicit space-time discontinuous Galerkin method on staggered unstructured meshes for the solution of the compressible Navier-Stokes equations at all Mach numbers, *J. Comp. Phys.*, vol. 341, pp. 341–376, 2017.

43. S. Boscarino, F. Filbet, and G. Russo, High Order Semi-implicit Schemes for Time Dependent Partial Differential Equations, *J. Sci. Comput.*, vol. 68, pp. 975–1001, 2016.

44. C.-W. Shu, High-order finite difference and finite volume weno schemes and discontinuous galerkin methods for cfd, *International Journal of Computational Fluid Dynamics*, vol. 17, no. 2, pp. 107–118, 2003.

45. Y. Saad and M. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 856–869, 1986.

46. J. Bell, P. Coletta, and H. Glaz, A second-order projection method for the incompressible Navier-Stokes equations, *J. Comput. Phys.*, vol. 85, pp. 257–283, 1989.

47. M. Dumbser, I. Peshkov, E. Romenski, and O. Zanotti, High order ADER schemes for a unified first order hyperbolic formulation of continuum mechanics: Viscous heat-conducting fluids and elastic solids, *Journal of Computational Physics*, vol. 314, pp. 824–862, 2016.

48. W. Boscheri, M.Dumbser, M.Ioriatti, I.Peshkov, and E.Romenski, A structure-preserving staggered semi-implicit finite volume scheme for continuum mechanics, *Journal of Computational Physics*, vol. 424, p. 109866, 2021.

49. U. Ghia, K. N. Ghia, and C. T. Shin, High-Re solutions for incompressible flow using Navier-Stokes equations and multigrid method, *Journal of Computational Physics*, vol. 48, pp. 387–411, 1982.

50. V. Casulli, A semi-implicit numerical method for the free-surface Navier-Stokes equations, *Int. J. Numer. Methods Fluids*, vol. 74, pp. 605–622, 2014.

51. W. Boscheri, A space-time semi-Lagrangian advection scheme on staggered Voronoi meshes applied to free surface flows, *Computers & Fluids*, vol. 202, p. 104503, 2020.

52. M. Tavelli, W. Boscheri, G. Stradiotti, G. R. Pisaturo, and M. Righetti, A mass-conservative semi-implicit volume of fluid method for the navier-stokes equations with high order semi-lagrangian advection scheme, *Computers and Fluids*, p. 105443, 2022.