

# Segregated Runge-Kutta time integration of convection-stabilized mixed finite element schemes for wall-unresolved LES of incompressible flows

Oriol Colomés<sup>a,b</sup>, Santiago Badia<sup>a,c</sup>

<sup>a</sup>*Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Parc Mediterrani de la Tecnologia, UPC, Esteve Terradas 5, 08860 Castelldefels, Spain ({ocolomes,sbadia}@cimne.upc.edu).*

<sup>b</sup>*Department of Civil and Environmental Engineering, Duke University, Durham, NC 27708, United States.*

<sup>c</sup>*Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.*

---

## Abstract

In this work, we develop a high-performance numerical framework for the large eddy simulation (LES) of incompressible flows. The spatial discretization of the nonlinear system is carried out using mixed finite element (FE) schemes supplemented with symmetric projection stabilization of the convective term and a penalty term for the divergence constraint. These additional terms introduced at the discrete level have been proved to act as implicit LES models. In order to perform meaningful wall-unresolved simulations, we consider a weak imposition of the boundary conditions using a Nitsche's-type scheme, where the tangential component penalty term is designed to act as a wall law. Next, segregated Runge-Kutta (SRK) schemes (recently proposed by the authors for laminar flow problems) are applied to the LES simulation of turbulent flows. By the introduction of a penalty term on the trace of the acceleration, these methods exhibit excellent stability properties for both implicit and explicit treatment of the convective terms. SRK schemes are excellent for large-scale simulations, since they reduce the computational cost of the linear system solves by splitting velocity and pressure computations at the time integration level, leading to two uncoupled systems. The pressure system is a Darcy-type problem that can easily be preconditioned using a traditional block-preconditioning schemes that only requires a Poisson solver. At the end, only coercive systems have to be solved, which can be effectively preconditioned by multilevel domain decomposition schemes, which are both optimal and scalable. The framework is applied to the Taylor-Green and turbulent channel flow benchmarks in order to prove the accuracy of the convection-stabilized mixed FEs as LES models and SRK time integrators. The scalability of the preconditioning techniques (in space only) have also been proven for one step of the SRK scheme for the Taylor-Green flow using uniform meshes. Moreover, a turbulent flow around a NACA profile is solved to show the applicability of the proposed algorithms for a realistic problem.

*Keywords:* Large eddy simulation, turbulence, variational multiscale, wall models, Runge-Kutta

---

## 1. Introduction

The numerical simulation of turbulent flows is widely used for scientific purposes and highly demanded in the industry to solve a large amount of engineering problems. The algorithms employed in the Computational Fluid Dynamics (CFD) field are constantly evolving, adapting to the new trends, and tailoring to the continually changing computational requirements. The increasing computational power acquired with the new improvements on supercomputers also involve additional advances in the software able to be executed in such machines.

Simulating incompressible turbulent flows involve the resolution of multiple scales, both in space and time, becoming a really challenging numerical problem. Direct Numerical Simulations (DNS) of turbulent flows are used to capture the physical phenomena at all scales, even the smallest ones. This approach has the inconvenience that consumes a large amount of computational resources, being impractical for many real industrial problems, even exploiting supercomputers. A technique that is increasingly used and that saves a lot of computational cost is the Large Eddy Simulation (LES), which basically consist on separating the flow in a coarse scale and a fine one, simulating the coarser and modelling the finer [1]. In order to model the fine scales in a LES method, one can consider a physically based approach defined

through additional modelling terms at the continuous level, or a purely numerical approach that does not introduce any modification to the governing equations. This last numerical approach is commonly denoted as Implicit LES (ILES) (see, e.g., [2]).

The Variational MultiScale (VMS) method introduced by Hughes in [3, 4] is a framework to develop stable and accurate numerical approximations of partial differential equations, preventing numerical instabilities that arise when the standard Galerkin Finite Element (FE) method is used. The use of VMS method as an ILES method was firstly suggested in [5, 6, 7] and, since then, several VMS methods have been developed and used as ILES. A two scale VMS approach based on an orthogonal definition of the subscales space has been proposed in [8], and named Orthogonal SubScale (OSS) method. OSS-type methods do not require to include the full residual in the stabilization term to end up with stable and accurate algorithms, as pointed out in [9], where a term-by-term OSS stabilization was proposed. It involved one term for the pressure stability (since equal order interpolation was used) and another term for the convection stabilization. Finally, in [10], inf-sup stable mixed FEs have been considered with a symmetric projection stabilization of the convective term. We note that any orthogonal projector can be considered in this framework, like, e.g., local-type projections [11, 12]. A complete assessment of VMS-type solvers for LES of turbulent flows can be found in [13, 10]. Briefly, all methods considered in these analyses showed comparable levels of accuracy, being convection-stabilization mixed methods cheaper in terms of CPU cost.

It is a common approach in the CFD field to consider a strong imposition of the Dirichlet boundary conditions, i.e., to impose the solution at the nodes of the Dirichlet boundary. This approach may lead to inaccurate solutions in some situations, specially when LES methods are considered to simulate wall-bounded turbulent flows. LES methods are motivated to be used on under-resolved meshes, which are insufficient to capture the boundary layers that appear in wall-bounded flows. The effect of not capturing properly the solution at the boundary layer can affect the mean flow, resulting in imprecise simulations. Thus, intensive refinement is required at the boundary layers in order to have fully-resolved meshes, destroying the desired cost reduction and nature of these schemes. In order to overcome this issue, a wall law must be considered, e.g., using a weak imposition of the Dirichlet boundary conditions can be contemplated. A Nitsche-type weak imposition was considered in [14] and later improved with a wall law based approach for turbulent flows in [15]. Alternatively, Robin-type boundary conditions have been considered in [16] to weaken the no-penetration and no-slip boundary conditions. Other examples of using weak Dirichlet boundary conditions can be found in [17]. In [14, 15] the wall-normal component of the velocity on the Dirichlet boundary is imposed strongly. This is an approach that one want to avoid due to its ill-posed definition on curved boundaries, since the normal vector is not defined on nodes; we note that only box-type domains were considered in [14, 15]. In this paper we consider a Nitsche's type weak imposition of the wall-normal component using wall law techniques and a standard Nitsche imposition of the tangent component, see [18, 19].

As noticed before, the turbulent phenomena is characterized by having not only many spatial scales, but also a multiscale description in time. Then, the time discretization becomes an important issue when simulating this kind of flows. Many authors favor implicit time integration schemes to avoid the time step restriction given by the Courant-Friedrichs-Lewy (CFL) number. However, at high Reynolds number, the hyperbolic CFL number (given by the convective term) has to be kept of the order of the unity (see, e.g., [20]). Then, it is a common practice to consider a semi-implicit time integration scheme where only the convective term is treated explicit, avoiding the need to satisfy the stringent parabolic CFL condition. See, e.g., [21], where a fractional-step method with Runge-Kutta time integration is used. Another example of this type of time integration schemes can be found in [22], where a semi-implicit BDF time integration scheme is considered together with a VMS-LES spatial discretization approach.

The time integration of turbulent flows in practical applications almost always involves a fractional-step method that segregates velocity and pressure computations, due to the clear reduction in computational time. This way, one can avoid the monolithic indefinite system and split the original problem into smaller coercive problems. However, fractional-step methods limit the order of accuracy to second order; due to the fact that the pressure must be extrapolated in the momentum equation, higher order schemes are unstable (see [23]). On the other hand, high order monolithic solvers are computationally demanding. Even using efficient block-preconditioning techniques (see, e.g., [24]), one must solve velocity and pressure problems at every solver iteration, whereas these iterations are usually due to the pres-

sure problem [10]. In order to solve this situation, the Segregated Runge-Kutta (SRK) time integration schemes for the incompressible Navier-Stokes equations have recently been proposed in [25], where they were tested for laminar flows. These schemes are based on two main goals. First, the segregation of the velocity and pressure computations at the time integration level, without the need to perform additional fractional step techniques that spoil high orders of accuracy. Second, the preservation of the same order of accuracy for both velocities and pressures. In this work, we will assess these schemes for turbulent flows. Otherwise stated, we will favour the IMPLICIT-EXPLICIT (IMEX) version of the SRK scheme that consists on treating implicitly the diffusive term and explicitly the convective term.

The velocity-pressure segregation introduced by the SRK schemes leads to elasticity-type and Darcy-type problems. The first problem is coercive, whereas the second can straightforwardly be preconditioned using a simple block-preconditioning strategy that only involves mass and Poisson solvers [10]. As a result, the use of SRK schemes drastically simplifies the linear system solve, compared to the indefinite systems that arise from monolithic formulations. When trying to solve increasingly larger problems on supercomputers, one has to efficiently exploit increasing levels of parallelism, i.e., to build scalable solvers. Due to the coercive nature of the building blocks in SRK time integration, in this work we consider the use of Balancing Domain Decomposition by Constraints (BDDC) preconditioners, firstly introduced in [26], following the implementations described in [27, 28]. The performance of the SRK time integration combined with block-preconditioning and BDDC domain decomposition preconditioning will be analyzed in this work.

The main contributions of this paper are:

- The use of convection-stabilized mixed FEs combined with weakly enforced normal and tangent velocity traces using Nitsche’s type methods, following the lines in [14, 15, 18, 19], in order to maintain under-resolved meshes in the whole domain, reducing computational cost (Sect. 2)
- The application and assessment of the newly introduced SRK schemes to turbulent flows with ILES modelling and its extension to weak Dirichlet boundary conditions, in order to split the problem into segregated velocity and pressure solvers without reducing the order of accuracy in time (Sect. 3)
- The development of a highly scalable parallel framework for the ILES simulation of turbulent flows, combining SRK time integration, block preconditioning, and a highly scalable implementation of BDDC solvers (Sect. 4)
- The assessment of the resulting methods in the ILES simulation of turbulent incompressible flows (Sect. 5)

These four developments are in the line to develop a highly accurate high-performance computational framework for the LES simulation of incompressible turbulent flows. The article is organized as follows. In Sect. 2 the Navier-Stokes problem is stated and the weak imposition of Dirichlet boundary conditions is described. We also introduce a mixed FE formulation with convection stabilization through OSS in Sect. 2.3. The SRK method, and its peculiarities when weak boundary conditions are considered, is defined in Sect. 3. In Sect. 4 the block-preconditioning technique and the parallel solver are defined. Three different tests have been considered in the numerical results section (Sect. 5), the Taylor-Green vortex flow, the turbulent channel flow test, and the flow around an airfoil defined with the NACA 0012 geometry. Finally, some conclusions are pointed out in Sect. 6.

## 2. Convection-stabilized mixed FE schemes

### 2.1. Navier-Stokes equations

Let  $\Omega$  be a bounded domain of  $\mathbb{R}^d$ , where  $d = 2, 3$  is the number of space dimensions,  $\Gamma = \partial\Omega$  its boundary and  $(0, T]$  the time interval. The strong form of the transient Navier-Stokes problem consists of finding the velocity field  $\mathbf{u}$  and the pressure field  $p$  such that

$$\partial_t \mathbf{u} - \nu \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega \times (0, T], \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times (0, T], \quad (2)$$

with  $\mathbf{f}$  the force vector and  $\nu$  the kinematic viscosity. Hereafter bold characters will denote vectors and tensors.

Equations (1) and (2) need to be supplied with appropriate boundary and initial conditions. The boundary  $\Gamma$  is divided into the Dirichlet ( $\Gamma_D$ ) and the Neumann ( $\Gamma_N$ ) parts such that  $\Gamma_D \cup \Gamma_N = \Gamma$  and  $\Gamma_D \cap \Gamma_N = \emptyset$ . Then, the boundary and initial conditions can be written as

$$\mathbf{u} = \mathbf{u}_g \quad \text{on } \Gamma_D \times (0, T], \quad (3)$$

$$(-p\mathbf{I} + \nu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)) \cdot \mathbf{n} = \mathbf{q}_N \quad \text{on } \Gamma_N \times (0, T], \quad (4)$$

$$\mathbf{u}(x, 0) = \mathbf{u}_0(x) \quad \text{in } \Omega \times \{0\}, \quad (5)$$

$\mathbf{n}$  being the unit outward vector normal to  $\Gamma$ .

In order to derive the weak form of the problem (1)-(5), we define some notation used hereinafter. We denote by  $L^p(\Omega)$ ,  $1 \leq p < \infty$ , the spaces of functions such that their  $p$ -th power is absolutely integrable in  $\Omega$ . In particular, for the case in which  $p = 2$ , we have a Hilbert space with scalar product

$$(u, v)_\Omega \equiv (u, v) := \int_\Omega u(x) v(x) d\Omega \quad (6)$$

and induced norm  $\|u\|_{L^2(\Omega)} \equiv \|u\| = (u, u)^{1/2}$ . Abusing of the notation, the same symbol as in (6) will be used for the integral of the product of two functions, even if these are not in  $L^2(\Omega)$ , and both for scalar and vector fields. The space of functions whose distributional derivatives up to order  $m$  are in  $L^2(\Omega)$  are denoted by  $H^m(\Omega)$ . We will focus on the case of  $m = 1$ , which is also a Hilbert space.  $H_0^1(\Omega)$  is the set of functions in  $H^1(\Omega)$  that have zero trace on  $\Gamma$ . Furthermore, we denote by  $H^{-1}(\Omega)$  the topological dual of  $H_0^1(\Omega)$  and by  $\langle \cdot, \cdot \rangle$  the duality pairing between  $H^{-1}(\Omega)$  and  $H_0^1(\Omega)$ . Given a Banach space  $X$ ,  $L^p(0, T; X)$  is the Bochner space of time dependent functions such that their  $X$ -norm is in  $L^p(0, T)$ .

From equations (1)-(5) one can derive the weak form of the problem, which consists in finding  $[\mathbf{u}, p] \in \mathbf{L}^2(0, T; \mathcal{V}_0) \times L^1(0, T; \mathcal{Q}_0)$  such that

$$(\partial_t \mathbf{u}, \mathbf{v}) + B(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) = \langle \mathbf{f}, \mathbf{v} \rangle + \langle \mathbf{q}_N, \mathbf{v} \rangle_{\Gamma_N} \quad \forall \mathbf{v} \in \mathcal{V}_0, \quad \forall q \in \mathcal{Q}_0, \quad (7)$$

satisfying the initial condition (5) in a weak sense. Here  $\mathcal{V}_0 := \mathbf{H}_0^1(\Omega)$  and  $\mathcal{Q}_0 := L^2(\Omega)/\mathbb{R}$ . The form  $B(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q))$  is defined as

$$B(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) := \nu(\nabla\mathbf{u}, \nabla\mathbf{v}) + b(\mathbf{u}, \mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) + (q, \nabla \cdot \mathbf{u}) \quad (8)$$

with the trilinear form of the convective term  $b(\mathbf{u}, \mathbf{v}, \mathbf{w})$  defined in its skew-symmetric version

$$b(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \frac{1}{2}(\mathbf{u} \cdot \nabla\mathbf{v}, \mathbf{w}) - \frac{1}{2}(\mathbf{v}, \mathbf{u} \cdot \nabla\mathbf{w}) + \frac{1}{2}(\mathbf{v}, (\mathbf{u} \cdot \mathbf{n})\mathbf{w})_{\Gamma_N}. \quad (9)$$

## 2.2. Weak Dirichlet boundary conditions

There are some situations in which the strong imposition of the Dirichlet boundary conditions (3) may have negative effects on the simulation performance. That is the case, for instance, of LES methods for wall-bounded flow problems. It is well known that LES methods have several difficulties when trying to solve wall-bounded problems (see, e.g., [29, 30]). The issue of solving this kind of problems using a LES method is that the boundary layer that appear next to the wall cannot be captured properly using under-resolved meshes. Then, there is the need of using very thin elements next to the wall in order to have at least one point in the viscous sublayer ( $y^+ \sim 1$ ) (see [31]). Usually, this requirement is satisfied by the use of stretched meshes, with very thin elements on the boundary and coarser ones in the middle of the channel.

Despite that, in many engineering problems it is more important to capture properly the large-scale flow properties than the fine scales, focusing on the effect of the boundary layers on the mean flow instead of trying to solve properly the boundary layer itself. In this direction, an approach is to make use of the fact that the velocity profile at the boundary layer has been shown to have a relation to the wall-normal distance [31]. Knowing this relation, one can impose weak boundary conditions on the

Dirichlet boundary enforcing that the traction generated at that walls is the one given by an analytic expression. This approach was followed by Bazilevs and co-workers in [15], showing that weakly-imposed boundary conditions provide the same results than strongly-imposed ones when using stretched meshes, but the former improves significantly the accuracy when using uniform meshes.

When weak Dirichlet boundary conditions are used, the functional space of the test functions lose the property of having zero trace on  $\Gamma_D$ , then  $\mathbf{v} \in \mathcal{V}$  and  $q \in \mathcal{Q}$ . Therefore, we have to consider the terms that arise when we use integration by parts to get the weak form of the Navier-Stokes equation (7). That is to include the following terms to the bilinear form.

$$B_\Gamma(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) = B(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) - ((-p\mathbf{I} + 2\nu\nabla^s \mathbf{u}) \cdot \mathbf{n}, \mathbf{v})_{\Gamma_D} + \frac{1}{2}((\mathbf{u} \cdot \mathbf{n})\mathbf{u}, \mathbf{v})_{\Gamma_D}. \quad (10)$$

Assuming that  $\mathbf{u} \cdot \mathbf{n} = 0$ , the term that is in charge of enforcing a given traction on the wall can be written as

$$(\tau_w \mathbf{t}, \mathbf{v})_{\Gamma_D} = \left( \tau_w \frac{(\mathbf{u} - \mathbf{u}_g)}{\|\mathbf{u} - \mathbf{u}_g\|}, \mathbf{v} \right)_{\Gamma_D} = (\alpha_b (\mathbf{u} - \mathbf{u}_g), \mathbf{v})_{\Gamma_D}, \quad (11)$$

where  $\mathbf{t}$  denotes the normalized vector acting on the direction of the traction and  $\tau_w$  the wall shear stress magnitude. The term (11) is nothing else than a penalty term with a parameter

$$\alpha_b := \frac{\tau_w}{\|\mathbf{u} - \mathbf{u}_g\|}. \quad (12)$$

In addition to this term, we also consider other boundary terms that arise when the so called Nitsche's method is used (see [32, 33]). Then, the resulting bilinear form (equivalent to (8)) will read

$$B_{weak}(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) = B_\Gamma(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) + (\alpha_b (\mathbf{u} - \mathbf{u}_g), \mathbf{v})_\Gamma - ((q \cdot \mathbf{I} + 2\nu\nabla^s \mathbf{v}) \cdot \mathbf{n}, (\mathbf{u} - \mathbf{u}_g))_\Gamma. \quad (13)$$

Note that the sign of the pressure test function in the second term of (13) is changed to keep the skew-symmetry of the velocity-pressure blocks.

The definition of the traction parameter  $\tau_w$  comes from the minimization of the residual of Spalding equation

$$y^+ = f(u^+) = u^+ + e^{-\chi B} \left( e^{\chi u^+} - 1 - \chi u^+ - \frac{(\chi u^+)^2}{2} - \frac{(\chi u^+)^3}{6} \right), \quad (14)$$

being  $\chi = 0.4$ ,  $B = 5.5$ ,  $y^+ := \frac{y u_\tau}{\nu}$  the wall distance and  $u^+ := \frac{\|\mathbf{u}_h\|}{u_\tau}$  the mean flow velocity in non-dimensional wall units. In this case, the distance to the wall is approximated to be proportional to the wall-normal mesh size,  $y = \frac{h_b}{p^2 C_b}$ , and  $\tau_w = u_\tau^2$ . Here,  $h_b$  denotes the wall-normal mesh size computed at each quadrature point,  $p$  the order of interpolation at the FE, and  $C_b$  a positive algorithmic constant.

A deeper description of this method as well as the algorithm for computing the parameter  $\alpha_b$  can be found in [15]. However, in that work, the authors impose strongly the normal component of the velocity ( $\mathbf{u} \cdot \mathbf{n} = 0$  on  $\Gamma_D$ ), a thing we want to avoid in the current work. As noticed in the introduction of the present paper, the normal vector is not defined on the nodes that belong to curved boundaries. In [18] the normal component of the velocity on the wall is also enforced weakly, adding an additional term to account for the weak no-penetration condition. Following the same approach used in [19], we split the penalty term distinguishing between the normal and tangential counterparts.

Let us consider a function  $\mathbf{v} \in \mathcal{V}$ . We can split its normal and tangential components as follows

$$\mathbf{v} = (\mathbf{v} \cdot \mathbf{n})\mathbf{n} + (\mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n}) = (\mathbf{n} \otimes \mathbf{n})\mathbf{v} + (\mathbf{I} - \mathbf{n} \otimes \mathbf{n})\mathbf{v}. \quad (15)$$

Introducing the decomposition (15) into the penalty term  $(\alpha_b (\mathbf{u} - \mathbf{u}_g), \mathbf{v})_{\Gamma_D}$ , and considering two different penalty parameters for each component,  $\alpha_{b,n}$  and  $\alpha_{b,t}$ , we obtain the equivalent penalty terms

$$(\alpha_{b,n} (\mathbf{u} - \mathbf{u}_g), (\mathbf{n} \otimes \mathbf{n})\mathbf{v})_{\Gamma_D} + (\alpha_{b,t} (\mathbf{u} - \mathbf{u}_g), (\mathbf{I} - \mathbf{n} \otimes \mathbf{n})\mathbf{v})_{\Gamma_D}. \quad (16)$$

In this case, the normalized tangential vector appearing in (11) reads  $\mathbf{t} = \frac{(\mathbf{I} - \mathbf{n} \otimes \mathbf{n})(\mathbf{u} - \mathbf{u}_g)}{\|(\mathbf{I} - \mathbf{n} \otimes \mathbf{n})(\mathbf{u} - \mathbf{u}_g)\|}$ . We use a wall law based definition of the Nitsche parameter for the tangential component, whereas we define the wall-normal component of the Nitsche parameter to be proportional to the original expression in [14], leading to the following definitions

$$\alpha_{b,n} := \beta \frac{C_b \nu p^2}{h_b}, \quad (17)$$

$$\alpha_{b,t} := \frac{\tau_w}{\|(\mathbf{I} - \mathbf{n} \otimes \mathbf{n})(\mathbf{u} - \mathbf{u}_g)\|}. \quad (18)$$

In (17),  $\beta$  is a constant that we can tune to adjust the model. An assessment of the parameter selection is done in Sect. 5.2. In any case, a large enough value of  $\beta$  is needed for stability purposes when the symmetric version of Nitsche's (or interior penalty) method is used (see [34]). Replacing the penalty terms in (13) by the ones in (16), we get a new expression for the bilinear form

$$\begin{aligned} B_{weak}(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) &= B_\Gamma(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) \\ &+ (\alpha_{b,n}(\mathbf{u} - \mathbf{u}_g), (\mathbf{n} \otimes \mathbf{n})\mathbf{v})_{\Gamma_D} + (\alpha_{b,t}(\mathbf{u} - \mathbf{u}_g), (\mathbf{I} - \mathbf{n} \otimes \mathbf{n})\mathbf{v})_{\Gamma_D} \\ &- ((\mathbf{u} - \mathbf{u}_g), (q\mathbf{I} + 2\nu\nabla^s \mathbf{v}) \cdot \mathbf{n})_{\Gamma_D}. \end{aligned} \quad (19)$$

In order to improve the stability of the segregated time integration schemes proposed in Sect. 3, we add additional boundary terms to weakly enforce the trace of the time derivative  $\partial_t \mathbf{u}$  on  $\Gamma_D$ . This point is further justified in Remark 3.3. In this case, the bilinear form reads

$$\begin{aligned} B_{weak,\delta t}(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) &= B_{weak}(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) \\ &+ (\alpha_{b,\delta t,n}(\partial_t \mathbf{u} - \partial_t \mathbf{u}_g), (\mathbf{n} \otimes \mathbf{n})\mathbf{v})_{\Gamma_D} + (\alpha_{b,\delta t,t}(\partial_t \mathbf{u} - \partial_t \mathbf{u}_g), (\mathbf{I} - \mathbf{n} \otimes \mathbf{n})\mathbf{v})_{\Gamma_D}, \end{aligned} \quad (20)$$

with  $\alpha_{b,\delta t,n} := \delta t \alpha_{b,n}$  and  $\alpha_{b,\delta t,t} := \delta t \alpha_{b,t}$ . Let us note that all the terms in (20) are dimensionally consistent with this definition of the penalty parameters.

### 2.3. Mixed FE method with convection stabilization

In order to define the semi-discrete problem, we consider a FE partition  $\mathcal{T}_h$  of the domain  $\Omega$ , from which we construct the conforming FE spaces for the velocity field,  $\mathcal{V}_h \subset \mathcal{V}$ , and for the pressure field,  $\mathcal{Q}_h \subset \mathcal{Q}$ . The Galerkin FE problem equivalent to (7) consists in finding  $[\mathbf{u}_h, p_h] \in \mathbf{L}^2(0, T; \mathcal{V}_{g,h}) \times L^1(0, T; \mathcal{Q}_{0,h})$  such that

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + B(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = \langle \mathbf{f}, \mathbf{v}_h \rangle + \langle \mathbf{q}, \mathbf{v}_h \rangle_{\Gamma_N} \quad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}, \forall q_h \in \mathcal{Q}_{0,h} \quad (21)$$

with the initial conditions (5) satisfied in a weak sense.  $\mathcal{V}_{0,h}$  and  $\mathcal{Q}_{0,h}$  denote the set of functions belonging to  $\mathcal{V}_h$  and  $\mathcal{Q}_h$ , respectively, with zero trace on  $\Gamma_D$ . Moreover, the velocity field function space is defined as  $\mathcal{V}_{g,h} := \{\mathbf{v}_h \in \mathcal{V}_h : \mathbf{v}_h|_{\Gamma_D} = \mathbf{u}_g\}$ . On the other hand, when weak boundary conditions are considered, the problem results in finding  $[\mathbf{u}_h, p_h] \in \mathbf{L}^2(0, T; \mathcal{V}_h) \times L^1(0, T; \mathcal{Q}_h)$  such that

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + B_{weak,\delta t}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = \langle \mathbf{f}, \mathbf{v}_h \rangle + \langle \mathbf{q}, \mathbf{v}_h \rangle_{\Gamma_N} \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \forall q_h \in \mathcal{Q}_h. \quad (22)$$

Note that in this case  $\mathbf{u}_h, \mathbf{v}_h \in \mathcal{V}_h$  and  $p_h, q_h \in \mathcal{Q}_h$ . Henceforth, we will assume that in the case when Dirichlet boundary conditions are considered, i.e.,  $\Gamma_D \neq \emptyset$ , they will be imposed weakly, making use of the bilinear form defined in (20).

Following the approach given by Codina in [9], we can build a VMS method with orthogonal definition of the subscales in which only the essential terms to guarantee stability are added to the Galerkin FE formulation. That is to consider the orthogonal projection of the convective term and the pressure gradient term separately. This approach reads: find  $[\mathbf{u}_h, p_h] \in \mathbf{L}^2(0, T; \mathcal{V}_h) \times L^1(0, T; \mathcal{Q}_h)$  such that

$$\begin{aligned} (\partial_t \mathbf{u}_h, \mathbf{v}_h) + B_{weak,\delta t}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) &+ (\tau_m \Pi_h^\perp(\mathbf{u}_h \cdot \nabla \mathbf{u}_h), \mathbf{u}_h \cdot \nabla \mathbf{v}_h)_h \\ &+ (\tau_m \Pi_h^\perp(\nabla p_h), \nabla q_h)_h + (\tau_c \nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)_h = \langle \mathbf{f}, \mathbf{v}_h \rangle \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \forall q_h \in \mathcal{Q}_h. \end{aligned} \quad (23)$$

In (23), the stabilization parameters  $\tau_m$  and  $\tau_c$  are usually defined as

$$\tau_m = \left( \frac{c_1 \nu}{h^2} + \frac{c_2 |\mathbf{a}|}{h} \right)^{-1}, \quad (24)$$

$$\tau_c = \frac{h^2}{c_1 \tau_m}, \quad (25)$$

where  $h$  is the mesh size and  $c_1$  and  $c_2$  are algorithmic constants. Alternatively to (25) one can consider the following definition of  $\tau_c$

$$\tau_c = c_c \left( \nu + \frac{c_1}{c_2} h |\mathbf{u}_h| \right), \quad (26)$$

which results from introducing (24) into (25) and considering a new constant  $c_c$ . This approach was analyzed in [10], where an implicit treatment of the projections was considered. Together with the static and linear version of the Algebraic Subgrid Scale (ASGS) method, the performance of the mentioned methods was tested for the LES of incompressible turbulent flows, and compared against convection-stabilized mixed FEs.

When inf-sup stable velocity-pressure FE pairs are used, the pressure gradient projection that appears in (23) can be neglected, since this term is not required for stability purposes.

After all this considerations, the final semi-discrete problem in which the current work is based consists in finding  $[\mathbf{u}_h, p_h] \in \mathbf{L}^2(0, T; \mathcal{V}_h) \times L^1(0, T; \mathcal{Q}_h)$  such that

$$\begin{aligned} (\partial_t \mathbf{u}_h, \mathbf{v}_h) + B_{weak, \delta t}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) + (\tau_m \mathbf{u}_h \cdot \nabla \mathbf{u}_h, \mathbf{u}_h \cdot \nabla \mathbf{v}_h)_h \\ - (\tau_m \boldsymbol{\eta}_h, \mathbf{u}_h \cdot \nabla \mathbf{v}_h)_h + (\tau_c \nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)_h = \langle \mathbf{f}, \mathbf{v}_h \rangle \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \forall q_h \in \mathcal{Q}_h, \end{aligned} \quad (27)$$

being  $\boldsymbol{\eta}_h := \Pi_h(\mathbf{u}_h \cdot \nabla \mathbf{u}_h)$  the projection of the convective term into the FE space, satisfying

$$(\tau_m \boldsymbol{\eta}_h, \mathbf{v}_h) = (\tau_m \mathbf{u}_h \cdot \nabla \mathbf{u}_h, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathcal{V}_h. \quad (28)$$

In (27), the last term of the left-hand side,  $(\tau_c \nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)_h$ , is usually called grad-div stabilization term, and in general it is not required to ensure stability of the problem. Nevertheless, when a stable mixed FE discretization is used, the influence of the grad-div stabilization on the accuracy of the method is essential. It has been shown that this term has an influence on the mass conservation [35], but it also influences the computational cost of the linear solvers [36, 37]. In [10], the effect of this term is assessed for the Taylor-Green Vortex problem and the Turbulent Channel Flow test using convection-stabilized mixed FE methods.

Let us finally note that other convection-stabilization methods can be used instead of OSS, which uses a  $L^2$  projector. Instead, local projection operators  $\boldsymbol{\kappa}_h^\perp(\cdot)$  can be considered [11, 12]. For an arbitrary symmetric projection stabilization of the convective term, the method reads:

$$\begin{aligned} (\partial_t \mathbf{u}_h, \mathbf{v}_h) + B_{weak, \delta t}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) + (\tau_m \boldsymbol{\kappa}_h^\perp(\mathbf{u}_h \cdot \nabla \mathbf{u}_h), \boldsymbol{\kappa}_h^\perp(\mathbf{u}_h \cdot \nabla \mathbf{v}_h))_h \\ + (\tau_c \nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)_h = \langle \mathbf{f}, \mathbf{v}_h \rangle \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \forall q_h \in \mathcal{Q}_h, \end{aligned}$$

### 3. Segregated Runge-Kutta time integration

#### 3.1. Matrix form

Let us consider the convection-stabilized mixed FE scheme applied to the Galerkin FE problem (21) (with strong imposition of boundary conditions). (It is obtained by replacing the bilinear form (20) by (8) in the definition of the stabilized problem (27).)  $\mathbf{u}_h$ ,  $p_h$  and  $\boldsymbol{\eta}_h$  are defined by their nodal values  $\{\mathbf{U}^a\}_{a=1, \dots, N_u}$ ,  $\{P^b\}_{b=1, \dots, N_p}$  and  $\{\boldsymbol{\Upsilon}^c\}_{c=1, \dots, N_\eta}$ , as

$$\mathbf{u}_h(x) = \sum_{a=1}^{N_u} \phi_a(x) \mathbf{U}^a, \quad p_h(x) = \sum_{b=1}^{N_p} \psi_b(x) P^b, \quad \boldsymbol{\eta}_h(x) = \sum_{c=1}^{N_\eta} \phi_\eta(x) \boldsymbol{\Upsilon}^c,$$

where  $\{\phi_{a,i}\}_{a=1,\dots,N_u;i=1,\dots,d}$ ,  $\{\psi_b\}_{b=1,\dots,N_p}$  and  $\{\phi_{c,i}\}_{c=1,\dots,N_\eta;i=1,\dots,d}$  are the Langrangian bases associated to  $\mathcal{V}_h$  and  $\mathcal{Q}_h$ .  $N_u$ ,  $N_p$  and  $N_\eta$  are the total amount of nodes for the velocity, pressure and projection fields. The matrix form of the stabilized problem with strong Dirichlet conditions can be written as follows:

$$\begin{bmatrix} \mathbb{M}\dot{\mathbf{U}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbb{K} + \mathbb{C} + \mathbb{A}_\tau & \mathbb{G} & \mathbb{B}_\tau \\ & \mathbb{D} & \mathbf{0} \\ -\mathbb{B}_\tau^T & \mathbf{0} & \mathbb{M}_\tau \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \\ \mathbf{\Upsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_u \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (29)$$

with  $\mathbb{M}^{ab} := (\phi_a, \phi_b)$ ,  $\mathbb{K}^{ab} := 2\nu(\nabla\phi_a, \nabla^s\phi_b)$ ,  $\mathbb{C}^{ab} := (\phi_a, \mathbf{u} \cdot \nabla\phi_b)$ ,  $\mathbb{A}_\tau^{ab} := \tau_m(\mathbf{u} \cdot \nabla\phi_a, \mathbf{u} \cdot \nabla\phi_b) + \tau_c(\nabla \cdot \phi_a, \nabla \cdot \phi_b)$ ,  $\mathbb{G}^{ab} := -(\nabla \cdot \phi_a, \psi_b)$ ,  $\mathbb{D}^{ab} := (\psi_a, \nabla \cdot \phi_b)$ ,  $\mathbb{B}_\tau^{ab} := -\tau_m(\mathbf{u} \cdot \nabla\phi_a, \phi_b)$ , and  $\mathbb{M}_\tau^{ab} := -\tau_m(\phi_a, \phi_b)$ , being  $a$  and  $b$  the node identification. Note that  $\mathbb{D} = -\mathbb{G}^T$ , when boundary conditions are considered.

### 3.2. Time integration through Segregated Runge-Kutta schemes

Focusing on the time integration of the problem (29) we consider now the application of a SRK scheme. SRK methods have been proposed in [25] and successfully tested for laminar flows. The main goal of this kind of time integrators is the segregation of velocity and pressure computations, giving systems much easier to be solved, taking advantage of an IMEX Runge-Kutta scheme. The use of a Rung-Kutta scheme also allows the use a high order integration in time with no extra implementation efforts. Furthermore, with this kind of time integrators, an adaptive time stepping technique can be easily used (see [38]).

Let us now consider the matrix problem (29), which can be written in a compact form in terms of the velocity and projection fields as follows

$$\mathbb{M}\dot{\mathbf{U}} = \mathcal{F}(\mathbf{U}) + \mathcal{G}(\mathbf{U}, \mathbf{\Upsilon}), \quad (30a)$$

$$-\mathbb{B}_\tau^T \mathbf{U} + \mathbb{M}_\tau \mathbf{\Upsilon} = \mathbf{0}. \quad (30b)$$

Where  $\mathcal{F}$  and  $\mathcal{G}$  are certain operators that contain the terms that will be integrated implicitly or explicitly, respectively. In particular we can define a SRK method in which the linear viscous term is integrated implicitly and the nonlinear terms, such as the convective and projection terms, and the pressure term are integrated explicitly. In this case, the operators  $\mathcal{F}$  and  $\mathcal{G}$  will read

$$\mathcal{F}(\mathbf{U}) := -\mathbb{K}\mathbf{U}, \quad (31)$$

$$\begin{aligned} \mathcal{G}(\mathbf{U}, \mathbf{\Upsilon}) := & \mathbf{F}_u - (\mathbb{C} + \mathbb{A}_\tau)\mathbf{U} - \mathbb{B}_\tau \mathbf{\Upsilon} \\ & + \mathbb{G}(\mathbb{D}\mathbb{M}^{-1}\mathbb{G})^{-1}(\mathbb{D}\mathbb{M}^{-1}((\mathbb{K} + \mathbb{C} + \mathbb{A}_\tau)\mathbf{U} + \mathbb{B}_\tau \mathbf{\Upsilon} - \mathbf{F}_u)). \end{aligned} \quad (32)$$

Note that this choice leads to a linear problem per time step since the only term integrated implicitly is the viscous term. Another fact to be highlighted here is that the projection and the velocity fields are not treated implicitly, making the use of symmetric projection stabilization schemes straightforward. Other choices could be used to define the operators  $\mathcal{F}$  and  $\mathcal{G}$ , e.g., the convective and the projection terms could be integrated implicitly. In this later case, the resulting system would be nonlinear and would couple velocity and projection fields. In general, we choose definitions (31)-(32) because we are interested in solving turbulent flows, where the time step restriction due to the hyperbolic CFL condition is often smaller than the restriction on the time step size required to capture the small time scales of the flow.

**Remark 3.1.** *The term  $(\tau_c \nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)_h$  integrated explicitly introduces a parabolic CFL condition, which is more restrictive than the hyperbolic one. Thus, when the matrix  $\mathbb{A}_\tau$  is treated explicitly we treat this term implicitly, with  $\tau_c$  evaluated at the previous time step. It does not spoil accuracy, since it is a stabilization term that is identically zero at the continuous level.*

We now proceed with the time integration scheme for equation (30a). Let us consider an IMEX RK method with  $s$  stages, the velocity and projection at the stage  $i$ ,  $\mathbf{U}_i$  and  $\mathbf{\Upsilon}_i$ , for  $1 \leq i \leq s$  are computed

as

$$\frac{1}{\delta t} \mathbb{M} \mathbf{U}_i = \frac{1}{\delta t} \mathbb{M} \mathbf{U}_n + \sum_{j=1}^i a_{ij} \mathcal{F}(\mathbf{U}_j) + \sum_{j=1}^{i-1} \hat{a}_{ij} \mathcal{G}(\mathbf{U}_j, \boldsymbol{\Upsilon}_j), \quad (33a)$$

$$- \mathbb{B}_\tau^T \mathbf{U}_i + \mathbb{M}_\tau \boldsymbol{\Upsilon}_i = \mathbf{0}. \quad (33b)$$

where  $a_{ij}$  and  $\hat{a}_{ij}$  are the coefficients of the implicit and explicit Butcher tableau, respectively. After some manipulation, we can rewrite (33) as

$$\frac{1}{\delta t} \mathbb{M} \mathbf{U}_i = \frac{1}{\delta t} \mathbb{M} \mathbf{U}_n + \sum_{j=1}^i a_{ij} \mathcal{F}(\mathbf{U}_j) + \sum_{j=1}^{i-1} \hat{a}_{ij} \mathcal{G}(\mathbf{U}_j, \mathbf{P}_j, \boldsymbol{\Upsilon}_j), \quad (34a)$$

$$- \mathbb{B}_\tau^T \mathbf{U}_i + \mathbb{M}_\tau \boldsymbol{\Upsilon}_i = \mathbf{0}, \quad (34b)$$

$$- \mathbb{D} \mathbb{M}^{-1} \mathbb{G}(\mathbf{P}_i) = \mathbb{D} \mathbb{M}^{-1} ((\mathbb{K} + \mathbb{C} + \mathbb{A}_\tau) \mathbf{U}_i + \mathbb{B}_\tau \boldsymbol{\Upsilon}_i - \mathbf{F}_u(t_i)). \quad (34c)$$

Let  $\mathcal{G}(\mathbf{U}, \mathbf{P}, \boldsymbol{\Upsilon}) := \mathbf{F}_u - (\mathbb{C} + \mathbb{A}_\tau) \mathbf{U} - \mathbb{B}_\tau \boldsymbol{\Upsilon} - \mathbb{G} \mathbf{P}$ , i.e., the operator equivalent to the one defined in (32) when the pressure variable is introduced. The final update after the  $s$  stages will read as follows

$$\frac{1}{\delta t} \mathbb{M} \mathbf{U}_{n+1} = \frac{1}{\delta t} \mathbb{M} \mathbf{U}_n + \sum_{i=1}^s b_i \mathcal{F}(\mathbf{U}_i) + \hat{b}_i \mathcal{G}(\mathbf{U}_i, \mathbf{P}_i, \boldsymbol{\Upsilon}_i), \quad (35a)$$

$$- \mathbb{B}_\tau^T \mathbf{U}_{n+1} + \mathbb{M}_\tau \boldsymbol{\Upsilon}_{n+1} = \mathbf{0}, \quad (35b)$$

$$- \mathbb{D} \mathbb{M}^{-1} \mathbb{G}(\mathbf{P}_{n+1}) = \mathbb{D} \mathbb{M}^{-1} ((\mathbb{K} + \mathbb{C} + \mathbb{A}_\tau) \mathbf{U}_{n+1} + \mathbb{B}_\tau \boldsymbol{\Upsilon}_{n+1} - \mathbf{F}_u(t_{n+1})). \quad (35c)$$

Note that equations (34c) and (35c) are equivalent to solve a Darcy type problem, with the following expression

$$\begin{bmatrix} \mathbb{M} & \mathbb{G} \\ \mathbb{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}^* \\ \mathbf{P}_k \end{bmatrix} = \begin{bmatrix} \mathbf{F}_u(t_k) - (\mathbb{K} + \mathbb{C} + \mathbb{A}_\tau) \mathbf{U}_k - \mathbb{B}_\tau \boldsymbol{\Upsilon}_k \\ 0 \end{bmatrix}, \quad (36)$$

with  $k$  being  $i$  or  $n+1$ , depending on the equation that we are solving, and  $\mathbf{U}^*$  an auxiliary velocity field, which satisfies the discrete incompressibility constraint. Then, in practice, equations (34c) and (35c) are implemented as a Darcy problem (36), which can be efficiently solved through block preconditioning technique (see Sect. 4).

**Remark 3.2.** *We note that SRK schemes cannot be used with pressure stabilization terms, i.e., typical VMS schemes that involve full residual stabilization and OSS schemes with equal-order velocity-pressure pairs. The reason is that SRK schemes have been designed for index-2 algebraic differential equations (as the incompressible Navier-Stokes equations) and these numerical schemes destroy the mathematical structure of the problem. Thus, stable mixed FEs are required when used with SRK schemes.*

### 3.3. Weak Dirichlet boundary conditions for SRK schemes

The use of weak boundary conditions is favoured when dealing with curved boundaries. In this case, the SRK time integration schemes have to take into account the boundary terms described in (10), (19) and (20). The algebraic form of (27)-(28), where weak Dirichlet boundary conditions are used, reads as follows:

$$\begin{bmatrix} (\mathbb{M} + \dot{\mathbb{M}}_\Gamma) \dot{\mathbf{U}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbb{K} + \mathbb{K}_\Gamma + \mathbb{C} + \mathbb{C}_\Gamma + \mathbb{A}_\tau + \mathbb{M}_\Gamma & \mathbb{G} + \mathbb{G}_\Gamma & \mathbb{B}_\tau \\ \mathbb{D} + \mathbb{D}_\Gamma & 0 & 0 \\ -\mathbb{B}_\tau^T & 0 & \mathbb{M}_\tau \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \\ \boldsymbol{\Upsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_u + \mathbf{F}_{u,\Gamma} + \dot{\mathbf{F}}_{u,\Gamma} \\ \mathbf{F}_{p,\Gamma} \\ \mathbf{0} \end{bmatrix}, \quad (37)$$

with the additional terms  $\dot{\mathbb{M}}_\Gamma^{ab} := (\alpha_{b,\delta t,n}(\mathbf{n} \otimes \mathbf{n}) \phi_a, \phi_b)_{\Gamma_D} + (\alpha_{b,\delta t,t}(\mathbf{I} - \mathbf{n} \otimes \mathbf{n}) \phi_a, \phi_b)_{\Gamma_D}$ ,  $\mathbb{M}_\Gamma^{ab} := (\alpha_{b,n}(\mathbf{n} \otimes \mathbf{n}) \phi_a, \phi_b)_{\Gamma_D} + (\alpha_{b,t}(\mathbf{I} - \mathbf{n} \otimes \mathbf{n}) \phi_a, \phi_b)_{\Gamma_D}$ ,  $\mathbb{K}_\Gamma^{ab} := -2\nu(\phi_a, \nabla^s \phi_b \cdot \mathbf{n})_\Gamma - 2\nu(\nabla^s \phi_a \cdot \mathbf{n}, \phi_b)_{\Gamma_D}$ ,  $\mathbb{C}_\Gamma^{ab} := \frac{1}{2}(\phi_a, (\mathbf{u} \cdot \mathbf{n}) \phi_b)_\Gamma$ ,  $\mathbb{G}_\Gamma^{ab} := (\phi_a, \psi_b \mathbb{I} \cdot \mathbf{n})_\Gamma$ ,  $\mathbb{D}_\Gamma^{ab} := -(\psi_a \mathbb{I} \cdot \mathbf{n}, \phi_b)_{\Gamma_D}$ ,  $\mathbf{F}_{u,\Gamma}^a := \alpha_b(\phi_a, \mathbf{u}_g)_\Gamma - 2\nu(\nabla^s \phi_a \cdot \mathbf{n}, \mathbf{u}_g)_{\Gamma_D}$ ,  $\dot{\mathbf{F}}_{u,\Gamma}^a := (\alpha_{b,\delta t,n}(\mathbf{n} \otimes \mathbf{n}) \phi_a, \dot{\mathbf{u}}_g(t_k))_{\Gamma_D} + (\alpha_{b,\delta t,t}(\mathbf{I} - \mathbf{n} \otimes \mathbf{n}) \phi_a, \dot{\mathbf{u}}_g(t_k))_{\Gamma_D}$ ,  $\mathbf{F}_{p,\Gamma}^a := -(\psi_a \mathbb{I} \cdot \mathbf{n}, \mathbf{u}_g)_{\Gamma_D}$ , being

$a$  and  $b$  the node identification. The velocity-pressure block keeps its skew-symmetry since  $\mathbb{D}_\Gamma = -\mathbb{G}_\Gamma^T$ . Note that the incompressibility constrain equation in this case reads

$$\tilde{\mathbb{D}}\mathbf{U} = \mathbf{F}_{p,\Gamma}, \quad (38)$$

being  $\tilde{\mathbb{D}} := (\mathbb{D} + \mathbb{D}_\Gamma)$ . (We note that this matrix is time-independent, i.e.,  $\partial_t \tilde{\mathbb{D}} = 0$ .) If we derive (38) in time and we operate with the first row of (37), we have that

$$\tilde{\mathbb{D}}\tilde{\mathbb{M}}^{-1}\tilde{\mathbb{M}}\dot{\mathbf{U}} + \tilde{\mathbb{D}}\tilde{\mathbb{M}}^{-1} \left[ (\tilde{\mathbb{K}} + \tilde{\mathbb{C}} + \mathbb{A}_\tau + \mathbb{M}_\Gamma)\mathbf{U} + \tilde{\mathbb{G}}\mathbf{P} + \mathbb{B}_\tau\boldsymbol{\Upsilon} \right] = \tilde{\mathbb{D}}\tilde{\mathbb{M}}^{-1}\tilde{\mathbf{F}}_u, \quad (39)$$

with  $\tilde{\mathbb{M}} := (\mathbb{M} + \mathbb{M}_\Gamma)$ ,  $\tilde{\mathbb{K}} := (\mathbb{K} + \mathbb{K}_\Gamma)$ ,  $\tilde{\mathbb{C}} := (\mathbb{C} + \mathbb{C}_\Gamma)$ ,  $\tilde{\mathbb{G}} := (\mathbb{G} + \mathbb{G}_\Gamma)$  and  $\tilde{\mathbf{F}}_u := (\mathbf{F}_u + \mathbf{F}_{u,\Gamma} + \dot{\mathbf{F}}_{u,\Gamma})$ . Using the fact that  $\tilde{\mathbb{D}}\tilde{\mathbb{M}}^{-1}\tilde{\mathbb{M}}\dot{\mathbf{U}} = \dot{\mathbf{F}}_{p,\Gamma}$ , we get from (39) the pressure system

$$\tilde{\mathbb{D}}\tilde{\mathbb{M}}^{-1}\tilde{\mathbb{G}}\mathbf{P} = \tilde{\mathbb{D}}\tilde{\mathbb{M}}^{-1} \left[ \tilde{\mathbf{F}}_u - (\tilde{\mathbb{K}} + \tilde{\mathbb{C}} + \mathbb{A}_\tau + \mathbb{M}_\Gamma)\mathbf{U} - \mathbb{B}_\tau\boldsymbol{\Upsilon} \right] - \dot{\mathbf{F}}_{p,\Gamma}, \quad (40)$$

where  $\dot{\mathbf{F}}_{p,\Gamma}^a = -(\psi_a \mathbb{I} \cdot \mathbf{n}, \dot{\mathbf{u}}_g)_\Gamma$ , for  $a = 1, \dots, N_p$ . Thus, when using weak Dirichlet boundary conditions in SRK time integration schemes, the Darcy type problem equivalent to (36) to be solved is

$$\begin{bmatrix} \tilde{\mathbb{M}} & \tilde{\mathbb{G}} \\ \tilde{\mathbb{D}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}^* \\ \mathbf{P}_k \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{F}}_u(t_k) - (\tilde{\mathbb{K}} + \tilde{\mathbb{C}} + \mathbb{A}_\tau + \mathbb{M}_\Gamma)\mathbf{U}_k - \mathbb{B}_\tau\boldsymbol{\Upsilon}_k \\ \dot{\mathbf{F}}_{p,\Gamma}(t_k) \end{bmatrix}, \quad (41)$$

with  $k$  corresponding to the stage  $i$  or the update  $n + 1$ .

**Remark 3.3.** Note that the auxiliary unknown  $\mathbf{U}^*$  in (41) is an approximation of  $\dot{\mathbf{U}}$ . When enforcing the Dirichlet data strongly, these conditions are straightforwardly applied to  $\mathbf{U}^*$ , since  $\mathbb{M}$  only involves free values. However, the trace of  $\mathbf{U}^*$  on  $\Gamma_D$  is not enforced in any sense when considering a weak imposition using (19), potentially leading to stability issues for segregated schemes. In order to solve this problem, we have included additional weak enforcement terms (over the acceleration) in (20) that are transferred to the pressure system through  $\tilde{\mathbb{M}}$ .

**Remark 3.4.** SRK schemes perturb the discrete mass conservation equation when time-dependent Dirichlet boundary conditions are enforced strongly. On the other hand, the discrete divergence constraint is exactly fulfilled in SRK schemes with weak imposition of Dirichlet data in all cases. This is another reason to favour the combination of SRK schemes and weak Dirichlet boundary conditions. See [25] for more details.

### 3.4. Adaptive time-stepping technique

One of the advantages when using a multi-stage time integration scheme, like a SRK method, is that an adaptive time stepping technique can be used straightforwardly. In fact, for a SRK scheme, the usage of an adaptive time stepping technique only requires an extra evaluation of the final velocity update, which involve a mass matrix solver, see [39, 40].

Our approach to define an adaptive time step is to follow the *PI11 controller* method described in [38] and suggested in [40], which computes the time step size as

$$\delta t_{n+1} = \left( \frac{\epsilon}{r_{n+1}} \right)^{1/k} \left( \frac{r_n}{r_{n+1}} \right)^{1/k} \frac{\delta t_n}{\delta t_{n+1}} \delta t_n,$$

with  $\epsilon = 0.8 \cdot TOL$ , where  $TOL$  is a given tolerance that we take as  $1 \cdot 10^{-5}$  and 0.8 is a safety factor. The local error is  $r_{n+1} = \|\mathbf{U} - \hat{\mathbf{U}}\|$  if the error per step (EPS) is controlled or  $r_{n+1} = \|(\mathbf{U} - \hat{\mathbf{U}})/\delta t_n\|$  if the error per unit step (EPUS) is controlled. In the former case  $k = p + 1$  (EPS) and for the second one (EPUS)  $k = p$ ,  $p$  being the order of the time integration scheme that has been used to compute the estimated velocity  $\hat{\mathbf{U}}$ . In this work we will favour the EPS definition.

Moreover, when treating with explicit versions of the SRK scheme, we have to limit the time step size to guarantee that the CFL condition is satisfied. When either the parabolic CFL condition,  $CFL_\nu = \frac{\nu \delta t}{(h/p^2)^2}$ , or the hyperbolic CFL condition,  $CFL_u = \frac{u \delta t}{(h/p^2)}$ , exceed the bound 1.0, the time step is rejected and reduced according to the safety factor  $\epsilon$ ,  $\delta t_{n+1, \text{new}} = \epsilon \delta t_{n+1, \text{old}}$ .

#### 4. Scalable solvers for incompressible Navier-Stokes equations

In this section we will briefly discuss the construction of efficient and scalable solvers for the linear systems arising from the space-time discretization of the Navier-Stokes equations in Sect.s 2-3. First of all we will have a look at the block preconditioning technique used in the different systems involved in the SRK scheme. Next, we will consider BDDC preconditioners to extract concurrency to be exploited in parallel contexts.

Note that if an IMEX version of the SRK scheme is considered, the resulting system from (30) can be solved sequentially, i.e., first solve the velocity block with explicit projections and then compute the projections for the following stage. Focusing on the Darcy-type equation (41), we consider an LU factorization of the form

$$\mathbb{A}_p := \begin{bmatrix} \tilde{\mathbb{M}} & \tilde{\mathbb{G}} \\ \tilde{\mathbb{D}} & 0 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0 \\ \tilde{\mathbb{D}}\tilde{\mathbb{M}}^{-1} & \mathbb{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbb{M}} & \tilde{\mathbb{G}} \\ 0 & \mathbb{S}_p \end{bmatrix}, \quad (42)$$

being  $\mathbb{S}_p := -\tilde{\mathbb{D}}\tilde{\mathbb{M}}^{-1}\tilde{\mathbb{G}}$  the Schur complement with respect to  $\mathbf{P}$ . In this case we consider a block-preconditioner based on an inexact LU factorization of  $\mathbb{A}_p$ , determined by the upper triangular matrix,

$$P(\mathbb{A}_p) := \begin{bmatrix} \tilde{\mathbb{M}} & \tilde{\mathbb{G}} \\ 0 & \mathbb{S}_p \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbb{M}}^{-1} & -\tilde{\mathbb{M}}^{-1}\tilde{\mathbb{G}}\mathbb{S}_p^{-1} \\ 0 & \mathbb{S}_p^{-1} \end{bmatrix}. \quad (43)$$

The solution of system (36) with a preconditioned GMRES method involve the computation of the inverse of matrices  $\tilde{\mathbb{M}}$  and  $\mathbb{S}_p$ . The approach followed to approximate  $\tilde{\mathbb{M}}^{-1}$  is the same we have defined for  $\mathbb{M}_\tau^{-1}$ , with one application of a diagonal DD preconditioner. The inverse of the Schur complement  $\mathbb{S}_p^{-1}$  is approximated by one application of a BDDC preconditioner over  $L_p$ . The fact that this choice is optimal (i.e., the two operators are spectrally equivalent) is straightforward, using the fact that  $\mathbb{S}_p$  and  $L_p$  are equivalent (see, e.g., [41]) and the fact that the BDDC preconditioner is optimal and scalable (see, e.g., [42]). The matrix systems arising from equations (35a) and (35b) are solved using a preconditioned GMRES with a diagonal DD preconditioner to approximate  $\mathbb{M}^{-1}$  and  $\mathbb{M}_\tau^{-1}$ , respectively.

The BDDC method, firstly introduced by Dohrmann [26], is a domain decomposition method, thus relying on a decomposition of the FE mesh into a set of subdomains. One of the main properties of BDDC is the possibility to extend the algorithm with a multilevel approach, based on the FE-type structure of the coarse matrix pattern (see, e.g., [43]). Regarding this property, a fully-distributed, communicator-aware, recursive, and interlevel-overlapped message-passing implementation of the multilevel balancing domain decomposition by constraints (MLBDDC) preconditioner has been developed in [28]. In these works, excellent weak scalability results have been obtained for the resolution of 3D Laplacian and elasticity problems on structured and unstructured grids, showing excellent weak scalability up to half a million cores and tens of billions of unknowns. Fortunately, using an IMEX SRK scheme and the block-preconditioning technique commented above, these are the only solvers needed for the simulation of turbulent flows. As a result, BDDC methods and the implementation of the MLBDDC proposed in [28] can readily be used. In some situations the mesh partition may lead to singular coarse problems, especially for the Laplacian-type problem. This problem can be overcome by the use of a perturbed BDDC with the addition of a properly scaled mass matrix at the preconditioner, see [44]. This approach will be used in Sect. 5.3, where a non-structured mesh partition is used.

**Remark 4.1.** *The development of weakly scalable and optimal preconditioners for monolithic incompressible Navier-Stokes schemes is a complicated task. As far as we know, the development of provably scalable methods has been restricted so far to mixed FEs with discontinuous pressures (see [45]). As a result, they cannot be applied, e.g., with the Taylor-Hood FEs used in Sect. 5.*

**Remark 4.2.** *Note that the problem is solved in a parallel framework for the spatial domain, while the time integration is treated sequentially. Thus, we are only considering space parallelism, not exploiting time parallelism. When we talk about scalability of the method, we refer to scalable solvers at a given time step within a sequential time integration scheme. The construction of parallel solvers, both in space and time for nonlinear turbulent flows is an open issue and it is out of the scope of this paper. However, initial*

research on this direction for BDDC solvers can be found in [46]. This issue has very important practical limitations. E.g., in the NACA simulation, we require heavy time stepping till reaching statistically relevant outputs, which requires huge computational resources.

**Remark 4.3.** Note that the scalability results shown in Sect. 5.1.3 have been carried out on uniform meshes. In realistic applications, meshes are usually unstructured and the preconditioning of the problem becomes more challenging, mainly due to the irregular decomposition of the domain. In any case, the preconditioners being used in this work have been shown to be scalable for unstructured meshes in [47].

## 5. Numerical tests

In this section we consider the numerical simulation of three popular incompressible turbulent flow benchmarks. The suitability of convection-stabilized inf-sup stable FEs with weak boundary conditions and SRK time integration schemes as an LES model for large scale problems is assessed in these tests. First, we analyze the advantages of using SRK methods in the TGV problem, where a weak scalability analysis is performed. The suitability of weak boundary conditions in wall-bounded flows is analyzed in the Turbulent Channel Flow test. Finally, some results are shown for the turbulent flow around an airfoil. It has been shown in [10] that convection-stabilized mixed methods have a great potential as LES models, especially when the computational cost is taken into account. This work aims to check the performance of these method when combined with SRK time integration schemes and a weak imposition of boundary conditions with a wall law. In both cases, the motivation is to reduce the computational cost of the simulations by 1) splitting the original indefinite problem into smaller coercive systems and 2) to use under-resolved meshes on boundary layers.

### 5.1. Taylor-Green vortex flow

The Taylor-Green Vortex (TGV) is a widely used benchmark for turbulent numerical simulations. It consists in a free-evolving flow starting from an analytic solution, in the domain  $(0, 2\pi)^3$ , with periodical boundary conditions (see, e.g., [48]). The initial condition is given by the following expression

$$\mathbf{u}(x, y, z, 0) = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \begin{pmatrix} u_0 \cos(x) \sin(y) \sin(z) \\ -u_0 \sin(x) \cos(y) \sin(z) \\ 0 \end{pmatrix}, \quad (44)$$

$$p(x, y, z, 0) = p_0 + \frac{1}{16} (\cos(2x) + \cos(2y)) (\cos(2z) + 2),$$

with

$$u_0 = \frac{2}{\sqrt{3}} \sin\left(\gamma + \frac{2\pi}{3}\right).$$

We choose  $\gamma = 0$ , which gives the mean initial velocity  $u_0 = 1$ . We solve the TGV problem for the Reynolds number  $\text{Re} = 1600$ .

The problem is solved from  $t = 0.0$  to  $T = 10.0$  and the results are compared against a DNS by Brachet et al [48]. We discretize the domain using different choices of the number of elements and the order of approximation, with either  $Q2/Q1$  or  $Q3/Q2$  Taylor-Hood velocity-pressure pairs. We consider the same algorithmic parameters ( $c_1 = 12.0$ ,  $c_2 = 4.0$ , and  $c_c = 4.0$ ) as in [10] for the TGV test when using mixed FEs.

#### 5.1.1. Monolithic vs Segregated Runge-Kutta

First, we want to compare the results obtained with a monolithic approach, using a second order Crank-Nicolson time integration scheme, against the ones computed using the IMEX version of the SRK method. In the later case we use the (3-3) scheme (see [25]).

The results shown in Fig. 1 have been computed using two different discretizations, one with  $16^3$   $Q2/Q1$  FEs and the other one with  $32^3$   $Q2/Q1$  FEs. The energy evolution and the energy dissipation rate are depicted in Fig. 1(a) and 1(b), respectively. The two methods exhibit a very similar behavior. The differences become smaller when the spatial mesh is refined.

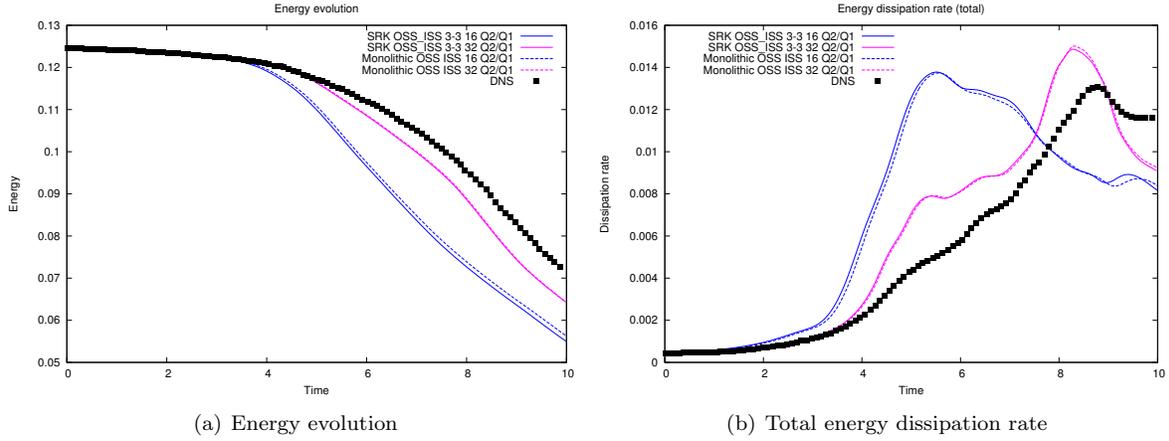


Figure 1: Energy and Total energy dissipation rate evolution with  $33^3$  velocity DOFs

### 5.1.2. *hp-refinement and adaptive time-stepping technique*

We clearly see that refining the mesh the results computed in Fig. 1 with convection-stabilized mixed FEs converge to the DNS. Moreover, it has been shown in [10] that a  $32^3$   $Q3/Q2$  elements mesh is fine enough to get very accurate results with these schemes.

One of the main problems when using explicit or IMEX time integration schemes is the restriction on the CFL number. In the case of the IMEX version of the SRK scheme, the diffusive term is treated implicitly and no restriction has to be satisfied for the parabolic CFL, but the convective term is treated explicitly. Then, we have to restrict the time step size in order to satisfy the hyperbolic CFL condition, i.e.,  $CFL_u \sim \delta t |\mathbf{u}| / (h/p^2) < 1$  everywhere. This restriction becomes more important when the mesh is refined, i. e.  $h$  is reduced, or  $p$  is increased.

An important reason for using Runge-Kutta schemes is the possibility to easily implement an adaptive time stepping technique, see Sect. 3.4. The important point of using this kind of techniques is that one can dynamically (and automatically) adapt the time step in order to satisfy both the physical and numerical requirements. The physical requirement on the time step size will be given by the change in the solution, while the numerical requirement will be given by the CFL condition.

In order to check the performance of the SRK method using adaptive time stepping, we solve the TGV problem with different mesh sizes and interpolations. In particular, we use  $16^3$  and  $32^3$  FE meshes with  $Q2/Q1$  and  $Q3/Q2$  discretizations, depicted in Fig 2. Looking at the kinetic energy evolution (Fig.

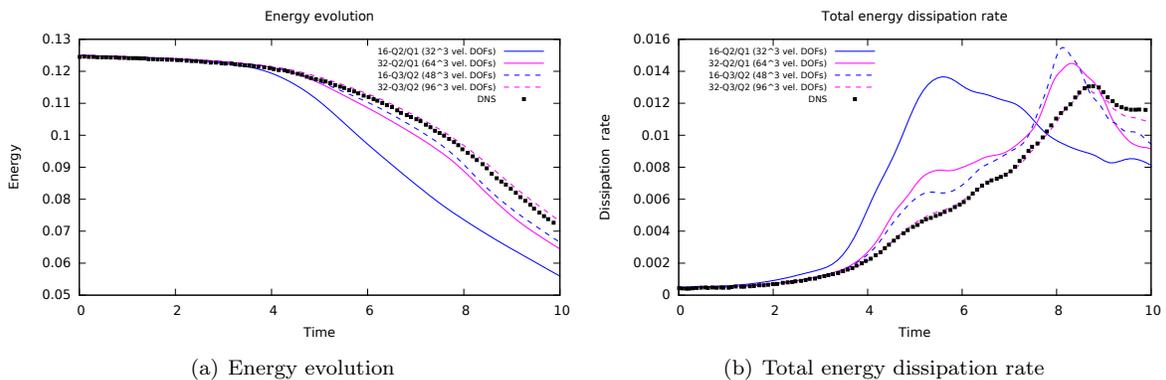


Figure 2: Energy and total energy dissipation rate evolution refining the mesh

2(a)) and at the total energy dissipation rate (Fig. 2(b)), we can clearly see that when the mesh is refined the solution tends to the DNS result. It is important to highlight here the improvement of the

results when higher order spatial approximation is used. We see that using  $Q3/Q2$  mixed FEs we get better results than using  $Q2/Q1$ , even with less DOFs. This behaviour is observed comparing the  $16^3$   $Q3/Q2$  elements mesh, with  $48^3$  velocity DOFs, against the  $32^3$   $Q2/Q1$  elements discretization, which have  $64^3$  velocity DOFs. It is seen that the former mesh results are in better agreement with the DNS solution.

As stated above, the mesh refinement has a direct implication in the time step size when explicit or IMEX schemes are used. To determine the influence of this refinement using an adaptive time stepping technique, we show the time step evolution for the four cases considered in this subsection, see Fig. 3. The initial time step is  $\delta t_0 = 5 \cdot 10^{-2}$  for all cases. It is seen that the coarser mesh ( $16^3$   $Q2/Q1$  elements) allows a higher time step, which is constantly increased until the difference between to time step solutions is too high. It occurs around  $t = 4.0$ , moment from which the time step size is adapted, remaining between 0.1 and 0.12. The  $16^3$   $Q3/Q2$  and the  $32^3$   $Q2/Q1$  show a similar behavior, with a first stage where the time step is increased until the CFL restriction is violated, point at which the time step is reduced, showing a sawtooth pattern from the beginning of the simulation. This pattern is repeated until the solution between two time steps is different enough to require a smaller time step to give more accurate solution. In what concerns the finest mesh,  $32^3$   $Q3/Q2$ , the CFL restriction prevails over the physical phenomena restriction. Consequently, the time step sawtooth pattern is observed during the whole simulation.

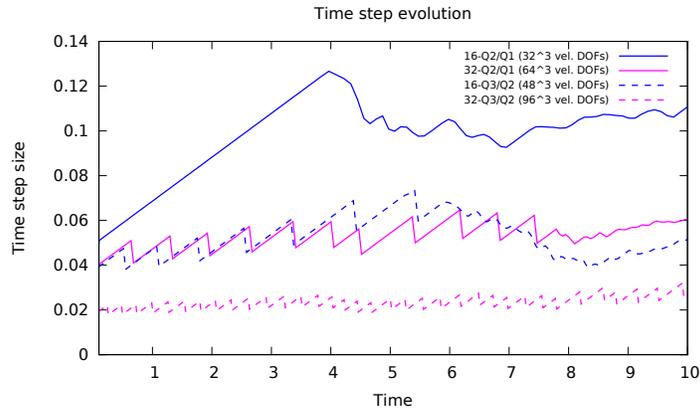


Figure 3: Time step size evolution refining the mesh

### 5.1.3. Scalability analysis

All the simulations have been performed with FEMPAR (Finite Element Multiphysics and massively PARallel) numerical software. FEMPAR is an open source, parallel hybrid OpenMP/MPI, object-oriented (OO) framework which, among other features, provides the basic tools for the efficient parallel distributed-memory implementation of substructuring domain decomposition solvers [49, 27, 28]. Under this parallel framework, an important issue is the scalability of the algorithms when large scale problems are solved. This means that, when refining the mesh, there is the need of having scalable solvers that do not increase the elapsed CPU time when the number of processors are increased. To check the suitability of the IMEX SRK algorithm for large scale problems we perform a weak scalability test solving the TGV problem.

Let us denote by  $H$  the characteristic length of the subdomain partition and  $L$  the one of the whole domain. We consider structured meshes and partitions of the cube  $[0, L]^3$ . In a weak scalability test we keep the local problems constant, i.e.,  $H/h$  fixed. In order to preserve the local Reynolds number (i.e., the relative importance of terms), we keep  $h$  and  $H$  constant and increase  $L$ ; we note that it implies to increase the number of FEs and subdomains (processors). Hence, as the reference problem is computed in a domain of size  $L_0 = 2\pi$ , the scalability analysis will be done in a domain of size  $L = \beta L_0$ , being  $\beta = \{1, 2, 3, 4, 5\}$ . The subdomain size is  $H = L_0/4$ . Let us note that the proposed local Reynolds number constant weak scalability analysis is much more challenging than typical weak

scalability analyses that reduce  $h$  and  $H$  keeping  $L$  constant. In the first analysis, the more processors we use, a more challenging physical problem is being solved (with a Reynolds number that increases linearly with  $L$ ), whereas in the second one the viscous terms are increasingly dominant, tending to fully resolved DNS simulations.

We compute the first time step of the Taylor-Green problem with time step size  $\delta t = 5 \cdot 10^{-2}$  using the IMEX version of the (1-2) scheme and  $Q2/Q1$  mixed FEs. We report the number of solver iterations and the CPU time consumed by each system resolution. Different local mesh sizes are considered to see the effect of the local mesh size on the scalability of the solver. In particular, we will consider three cases:  $H/h = \{4, 8, 12\}$ . The total amount of processors used to solve the problem will be  $(L/H)^3 = (4\beta)^3$ . The smallest mesh used to solve the problem in this analysis is a  $16^3$   $Q2/Q1$  FEs mesh, while the biggest one is a  $240^3$   $Q2/Q1$  FEs mesh, leading to more than 680 million DOFs. The solution is computed in the HLRN-III Cray system in Hannover (Germany). In particular, we use the Cray XC40 MPP2 system that is equipped with two Intel Xeon Haswell 12-CPU cores per node.

The following results are computed with a two-level BDDC solver, considering the corners, edges and faces of the subdomain partitions as part of the coarse system (see [49, 27, 28] for details). We recall that the first step of a SRK (1-2) scheme consists in six system resolutions that are specified in Algorithm 1. Note that the first system solver is only needed at the first step, since after that  $p_1 = p_n$ .

---

**Algorithm 1:** SRK system resolutions for one time step using the (1-2) scheme

---

1. Solve initial pressure (equation (34c)).
  2. Solve second-stage momentum system (equations (34a)-(34b)).
  3. Solve second-stage pressure (equation (34c)).
  4. Solve velocity update (equation (35a)).
  5. Solve projection update (equation (35b)).
  6. Solve pressure update (equation (35c)).
- 

Fig. 4 depicts the solver iterations and the elapsed CPU time for the constant local Reynolds number weak scalability analysis with the three different local mesh sizes. Solver iteration plots (Fig. 4(a), Fig. 4(c), and Fig. 4(e)) and CPU time plots (Fig. 4(b), Fig. 4(d), and Fig. 4(f)) are shown for the six linear systems; the pressure systems CPU times have been added together in the second set of plots. The building time includes the time consumed to integrate all matrices and vectors and the time consumed computing the preconditioners.

Looking at the solver iterations, we see that the number of iterations for each linear system solve is kept constant when the number of processors is increased, independently of the local mesh size. We notice that when the local mesh sizes increase, the number of iterations decrease a little for all systems. Further, we can observe at the solver iterations column in Fig. 4 that, as expected, the velocity update and the projection update give the same number of solver iterations. This is caused by the fact that both systems solve a mass matrix, with the only difference that the velocity update mass matrix is scaled by  $\frac{1}{\delta t}$ . We also observe the well-known fact that, in terms of solver iterations, the hardest system is the pressure solver (equations (34c) and (35c)).

In what refers to the consumed CPU time, a first conclusion we can take looking at Fig. 4(b), Fig. 4(d), and Fig. 4(f), is that the most expensive resolution is the momentum system (equations (34a)-(34b)). Although the number of solver iterations is smaller than the pressure computations, in terms of time consumed, the momentum equations resolution is more expensive, even counting together the three pressure solvers. This difference in the computational time required for the momentum equations versus the pressure ones, is due to the number of DOFs that are being solved. We have to take into account that both velocity and projection fields are computed with a second order polynomial,  $Q2$ , while the pressure is first order,  $Q1$ . Then, we not only have two vectorial fields versus a scalar field, but we have a second order interpolation versus a first order one. Leaving aside the mass matrix blocks, the size of the system of equations to be solved in the velocity block of the momentum computation is  $\sim 24$  times larger than the one that arise for the pressure block of the Darcy computations, taking into account that the pressure system is solved using (41).

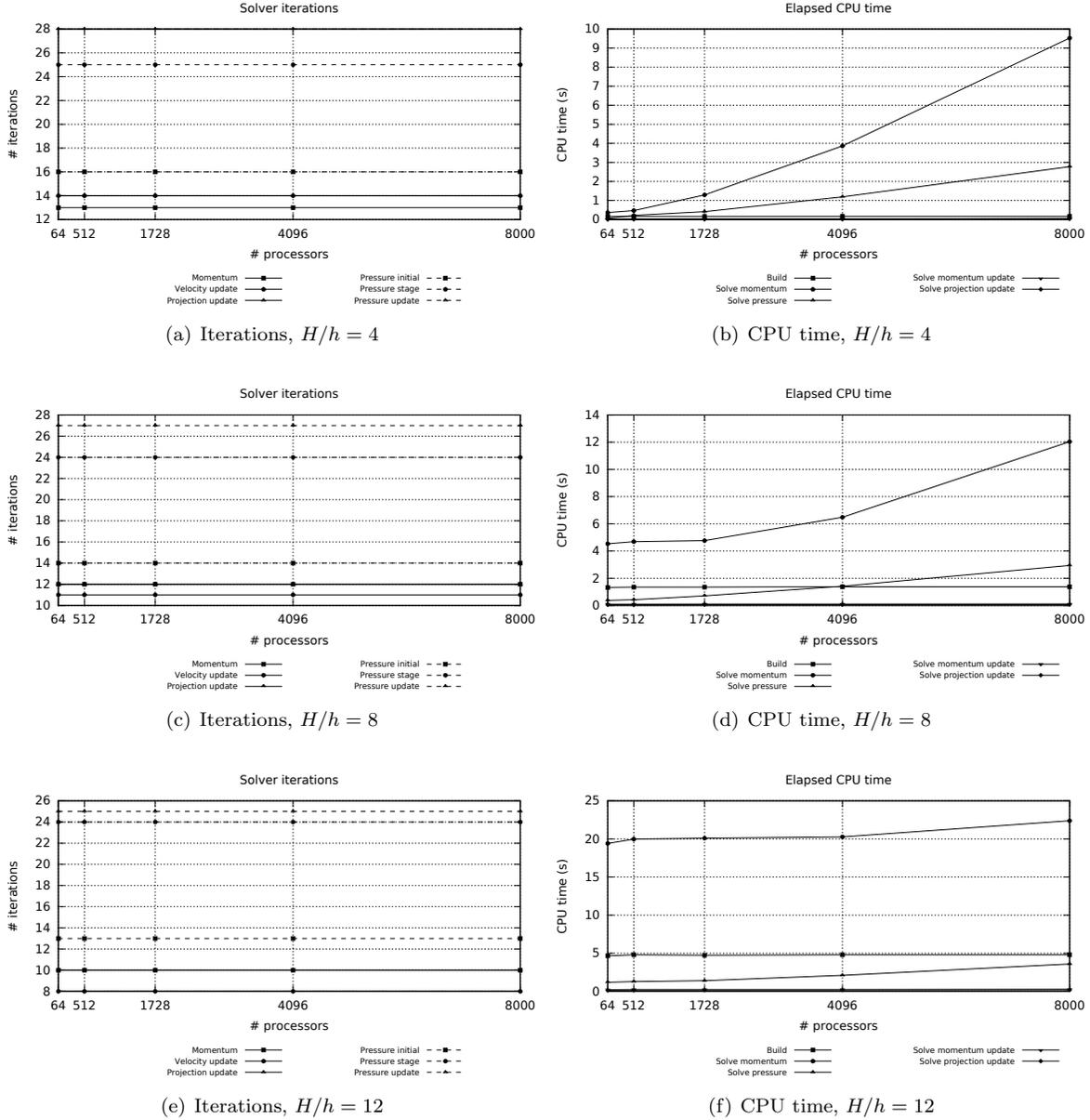


Figure 4: Weak scalability test. Number of iterations and elapsed CPU time for solving one time step of the TGV problem with different local mesh sizes.

Keeping with the discussion about the CPU time shown in Fig. 4, we see that the computations of the momentum and pressure systems do not scale in terms of time. This performance has an easy explanation, which is given by the ratio between the fine and coarse BSSC system sizes. The coarse system size only depends on the number of processors (subdomains) used by the BDDC method. Even though it is some orders of magnitude smaller than the fully parallel component, we have considered a serial solver for the coarse component. Further, the coarse problem size increases with the number of processors. Thus, at some point, the non-scalable coarse solver becomes dominant. When corners, edges and faces of the subdomain are considered, the total amount of degrees of freedom of the coarse system is equivalent to a local mesh of  $Q2$  elements (without the interior node) with as many elements as subdomains. Consequently, for a  $H/h = 4$  local mesh, we can only expect to have weak scalability until 64 ( $4^3$ ) processors, for  $H/h = 8$  until 512 ( $8^3$ ) processors, and for  $H/h = 12$  until 1728 ( $12^3$ )

processors. In fact this is what we are seeing in Fig. 4, where, taking into account that the local meshes have the interior DOF inside each element, the results show the expected scalability performance. In order to have even better results, one can reduce the size of the coarse system by considering only the corners and the edges of the subdomain. In Fig. 5 we depict the results obtained in this case, for a  $H/h = 12$  local mesh size. It is seen that when the subdomain faces are ignored, the number of solver iterations increase, see Fig. 5(a). Despite that, in the trade-off of having more solver iterations with smaller system of equations, ignoring the subdomain faces in the coarse system is worth. As we can see in Fig. 5(b), the momentum system is scalable up to 8.000 processors since the local problem is more expensive than the coarse one. We cannot scale in time for the pressure solver because the local problem is much smaller than the coarse as we use  $Q1$  elements for the pressure field. In any case, the

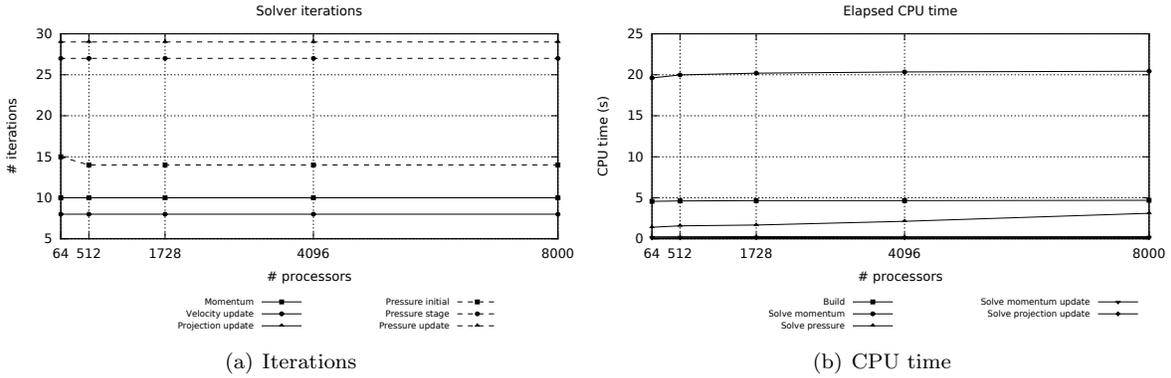


Figure 5: Weak scalability test. Number of iterations and elapsed CPU time for solving one time step of the TGV problem with  $H/h = 12$  and only corners and edges at the coarse level.

coarse problem can recursively be approximated by its BDDC preconditioner, leading to the multilevel BDDC method. It has been shown that the multilevel BDDC implementation within FEMPAR is able to scale up to 2 million subdomains and half a million MPI tasks by using three levels for the Laplacian problem and four levels for the elasticity problem [28]. We have not considered the multilevel extension here, since we have not considered these extreme scales.

## 5.2. Turbulent channel flow at $Re_\tau = 395$

In the previous section we have tested the suitability of the proposed VMS method using an IMEX SRK time integration scheme for the LES simulation of homogeneous turbulence. After that, we aim to show the performance of such methods for wall-bounded turbulent flows. In order to check the behaviour of the proposed approach, we solve the Turbulent Channel Flow (TCF) test with a Reynolds number based on the wall velocity  $Re_\tau = 395$ . One of the main purposes of this test is to assess the accuracy of the solution when using the weak boundary conditions formulation given by the bilinear form described in (19).

The TCF problem is solved in the boxed domain  $[0, 2\pi\delta] \times [0, 2\delta] \times [0, 2\pi\delta/3]$ , where the  $x$ -direction is the flow direction, also called stream-wise direction, the  $y$ -direction is the wall-normal direction, and the  $z$ -direction is the span-wise direction. Homogeneous Dirichlet boundary conditions for the velocity DOFs are weakly imposed on wall-normal direction boundaries ( $y = -\delta$  and  $y = \delta$ ), while periodic boundary conditions are defined on the stream-wise and span-wise directions. Fig. 6 depicts schematically the geometry of the computational domain for this test. We select the problem parameters according to the DNS computation performed by Moser et al in [50, 51] (MKM-DNS), so we can compare our results with the cited work. The bulk mean velocity and the half channel height are taken equal to one,  $\bar{U} = 1$  and  $\delta = 1$ . The estimated Reynolds number based on the bulk mean velocity is known,  $Re = \bar{U}2\delta/\nu \approx 13,750$  (see [31]). Therefore, one can obtain the value of the viscosity,  $\nu = 1.4545 \cdot 10^{-4}$  and determine the friction velocity magnitude from the Reynolds number based on the friction velocity as  $u_\tau = Re_\tau\nu/\delta = 5.745 \cdot 10^{-2}$ . Thus, the wall shear stress reads  $\tau_w = u_\tau^2 = 3.3010 \cdot 10^{-3}$ . A force

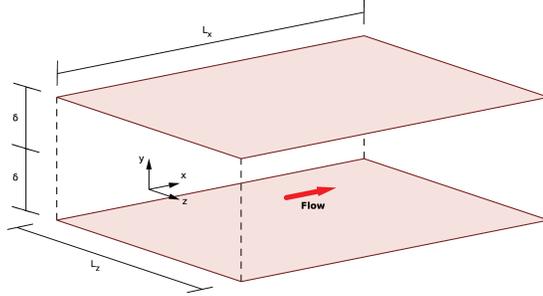


Figure 6: Channel computational domain.

equivalent to a pressure gradient is imposed to drive the movement of the flow in the stream-wise direction,  $f_x = \tau_w/\delta$ .

An initial solution proposed in [52] is used to reach the statistically steady state faster, which consists in a unidirectional velocity profile with a fluctuation:

$$\begin{aligned} u_x &= C(1 - y^8) + \epsilon \frac{L_x}{2} \sin(\pi y) \cos\left(\frac{4\pi x}{L_x}\right) \sin\left(\frac{2\pi z}{L_z}\right), \\ u_y &= -\epsilon(1 + \cos(\pi y)) \sin(\pi y) \sin\left(\frac{4\pi x}{L_x}\right) \sin\left(\frac{2\pi z}{L_z}\right), \\ u_z &= -\epsilon \frac{L_z}{2} \sin\left(\frac{4\pi x}{L_x}\right) \sin(\pi y) \cos\left(\frac{2\pi z}{L_z}\right). \end{aligned} \quad (45)$$

The constant  $C$  is chosen in a such way that the field without fluctuations would have a bulk mean velocity  $\bar{U} = 1.0$ . The fluctuation constant  $\epsilon$  is 10% of the bulk mean velocity.

To solve the problem we use different meshes, composed by  $16^3$   $Q2/Q1$  and  $Q3/Q2$  FEs. We consider both stretched meshes and wall-unresolved uniform meshes along the wall-normal direction. To construct the stretched mesh we follow the hyperbolic function defined in [13] with a stretching parameter  $\gamma = 2.75$ . According to [10], the algorithmic constants that appear in (24) and (26) are chosen as  $c_1 = 12.0$ ,  $c_2 = 8.0$  and  $c_c = 32.0$ . The constant that appear in the weak boundary conditions formulation is set to  $C_b = 32.0$ . Moreover, we will analyze the effect of the constant  $c_c$  when uniform meshes are considered, as it plays an important role when using mixed FE methods.

### 5.2.1. Effect of $c_c$ on uniform meshes

The effect of the constant  $c_c$  that appear in equation (26) have been studied in [10] for the case in which we have strong imposition of the boundary conditions together with the use of stretched meshes that capture the boundary layer. In this section we aim to check the performance of the method when changing the  $c_c$  algorithmic parameter in uniform meshes and weak Dirichlet boundary conditions. In this case, in order to avoid the influence of the weak imposition of the wall normal component, we enforce strongly the wall-normal velocity, e.g.  $\mathbf{u} \cdot \mathbf{n} = 0$  on  $\Gamma_D$ .

In Fig. 7 we show the energy evolution of the solution for different values of  $c_c$  from  $t = 0.0$  to  $t = 100.0$ , and computed on a  $16^3$   $Q2/Q1$  mesh. It can be shown that for uniform meshes, high values of  $c_c$  makes the solution more energy conserving, so the energy is not dissipating properly. The value  $c_c = 1.0$  is the first case that reaches the statistically stable state, at  $t \sim 280.0$ , with the lowest global energy. Note that the case  $c_c = 0.0$  results in an overdissipating method, showing the importance of adding the grad-div term, as noticed in [10]. Then, in following sections we will use  $c_c = 1.0$ .

### 5.2.2. Influence of the wall-normal component

A significant difference between the approach followed herein with respect to the one proposed in [15] is the treatment of the normal component. In [15] a no-penetration condition is considered, which means that the normal component of the velocity is imposed strongly,  $\mathbf{u} \cdot \mathbf{n} = 0$  on  $\Gamma_D$ . We propose to

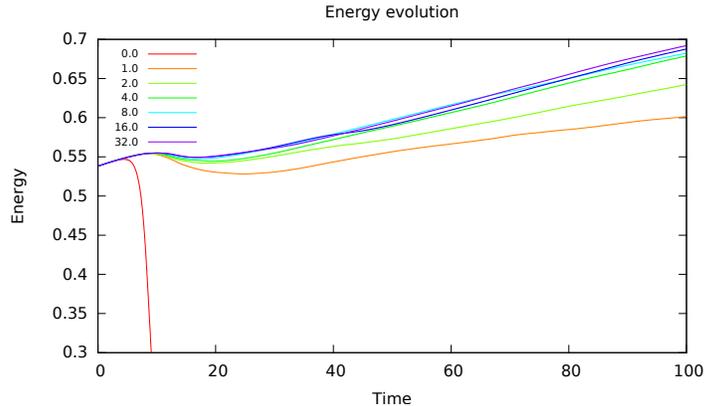


Figure 7: Energy evolution for the TCF test for different values of  $c_c$ .

keep the weak boundary imposition also in the wall-normal direction, due to the reasons commented above.

As stated in Sect. 2.2, we define two difference penalty parameters. One for the tangential component, based on the wall law, and another one for the wall-normal component, which is proportional to the original penalty parameter proposed in [14],  $\alpha_{b,n} := \beta \frac{C_b \nu p^2}{h_b}$ . As it is obvious, when  $\beta \rightarrow \infty$  the solution at the wall becomes closer to the one obtained imposing strongly the normal component, but the penalty term may disturb the solution. In order to determine the best value for  $\beta$ , we solve the TCF test from  $t = 0.0$  to  $t = 100.0$ , with different values  $\beta = \{1.0, 10, 100, 1000\}$  in a  $16^3$   $Q_2/Q_1$  FEs uniform mesh. The problem is solved using an IMEX SRK scheme with adaptive time stepping technique. In Fig. 8 we show the energy evolution for the different values of  $\beta$ . We clearly see that the choice of  $\beta$  has not a large implication on the results unless too low values of  $\beta$  are used. (It is well-known that stability of Nitsche boundary conditions can only be proved for *large enough* values of  $\beta$  (see [34]). For too low values of  $\beta$ , the case of  $\beta = 1.0$ , the solution starts dissipating too much energy at the early stages of the simulation. When increasing  $\beta$ , the results tend to the ones obtained considering a strong imposition of the normal component, denoted by *strong*. We will take  $\beta = 10.0$  in the following simulations.

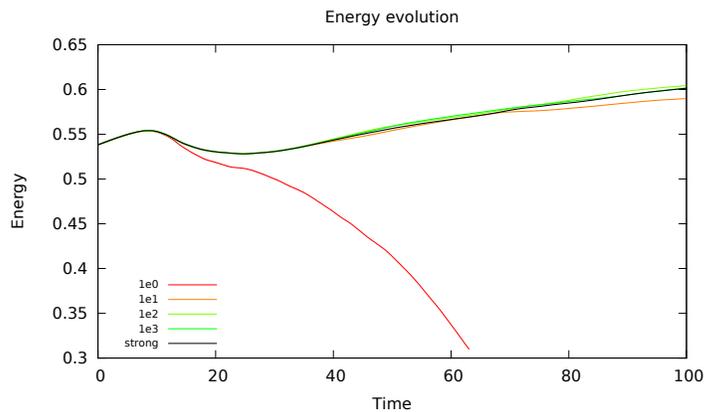


Figure 8: Energy evolution for the TCF test for different values of  $\beta$ .

### 5.2.3. Comparison between strong and weak imposition of Dirichlet boundary conditions

Once analyzed the effect of the  $\beta$  parameter when Dirichlet boundary conditions are imposed weakly for the wall-normal component, let us check the performance of the approach followed in this work compared against a strong Dirichlet boundary conditions definition. We will compare with the results

presented in [10], where the TCF test was solved using strong Dirichlet boundary conditions in a stretched mesh. In Table 1 we state the distance to the wall of the nearest velocity DOF for a stretched and a uniform mesh, with two different discretizations.

Table 1: Distance to the wall of the nearest velocity DOF in wall units.

Mesh type	Discretization	$y^+$
Stretched	$16^3$ $Q2/Q1$	1.6
	$16^3$ $Q3/Q2$	1.0
Uniform	$16^3$ $Q2/Q1$	24.7
	$16^3$ $Q3/Q2$	16.5

The element size given in Table 1 has a direct implication on the choice of the time integration scheme to be used. If we want to use an IMEX version of the SRK scheme we have to ensure that the hyperbolic CFL condition is satisfied. Consequently, with the stretched mesh we will have a stronger restriction than with the uniform mesh, for a given discretization. Thus, the fact that one needs a full resolution of the boundary layer does not only affect the spatial mesh but also a DNS-like time step size. For this particular test, using a uniform mesh with  $16^3$   $Q2/Q1$  elements, the critical time step size is around  $\delta t_{crit} = 0.025$ , while for the stretched mesh with the same discretization is around  $\delta t_{crit} = 0.007$ . Following [53], a good time step choice for this particular TCF when implicit time integration schemes are used is  $\delta t = 0.03$ . Taking into account that the IMEX version of the SRK schemes is about three time faster per time step than the implicit one, we have considered IMEX SRK for the uniform mesh and the implicit SRK scheme (with  $\delta t = 0 - 03$ ) for the stretched mesh. The TCF test is solved from  $t = 0.0$  until it reaches the statistically stable state, storing the statistics from the last 30.0 seconds, accumulating more than 1000 samples.

The results for the  $16^3$   $Q2/Q1$  elements mesh are shown in Fig. 9, where the mean stream-wise velocity is plotted (Fig. 9(a)) together with the velocity fluctuations in the stream-wise, wall-normal and span-wise directions (Figs. 9(b), 9(c) and 9(d), respectively). When a stretched mesh is used, we see that with weak boundary conditions imposition we recover the same results as the strong case. Similar results are observed for the velocity fluctuations. As expected, when using heavily under-resolved meshes on the boundary layers (see Table 1), the mean stream-wise velocity profile given by the uniform mesh with weak boundary conditions is a little bit far from the DNS results and also from the results obtained with strong boundary conditions in a stretched mesh (see Fig. 9(a)). Analogously, looking at Fig. 9(b), where the stream-wise velocity fluctuation is depicted, we see that the nearest to the wall point is far to the DNS curve for the weak and uniform case. However, far enough from the boundary layer, the results are very close to the DNS for a much coarser space-time mesh. The same behaviour is observed for the wall-normal and span-wise velocity fluctuations, Figures 9(c) and 9(d).

An improvement in the solution is observed when higher order discretization is used in space. This behaviour is shown in Fig. 10 where the uniform meshes with weak boundary conditions imposition is compared against the stretched mesh with strong boundary conditions, both of them with a  $16^3$   $Q2/Q1$  and a  $16^3$   $Q3/Q2$  elements mesh. Looking at the mean stream-wise velocity profile shown in Fig. 10(a), we see that solution converge to the DNS results as we refine the mesh. Same performance is observed in the velocity fluctuations. For instance, if we look at the stream-wise velocity fluctuation shown in Fig. 10(b) it is seen that only the nearest to the wall node is far from the DNS for the  $Q3/Q2$  discretization, while all the remaining points are very close to the DNS curve. In what refers to the wall-normal and span-wise velocity fluctuations depicted in Figures 10(c) and 10(d), respectively, we see that better results are obtained, specially on the points near to the wall.

It is important to highlight that 1) very similar results are obtained with strong and weak imposition of the Dirichlet boundary conditions on the same stretched meshes, and 2) that the weak imposition allows for under-resolved meshes with good agreement with the DNS far from the wall.

### 5.3. Flow around an airfoil at low Reynolds number

The aim of this section is to apply the numerical schemes in this work to the simulation of the turbulent incompressible flow around an airfoil, particularly a NACA 0012 airfoil [54]. The NACA

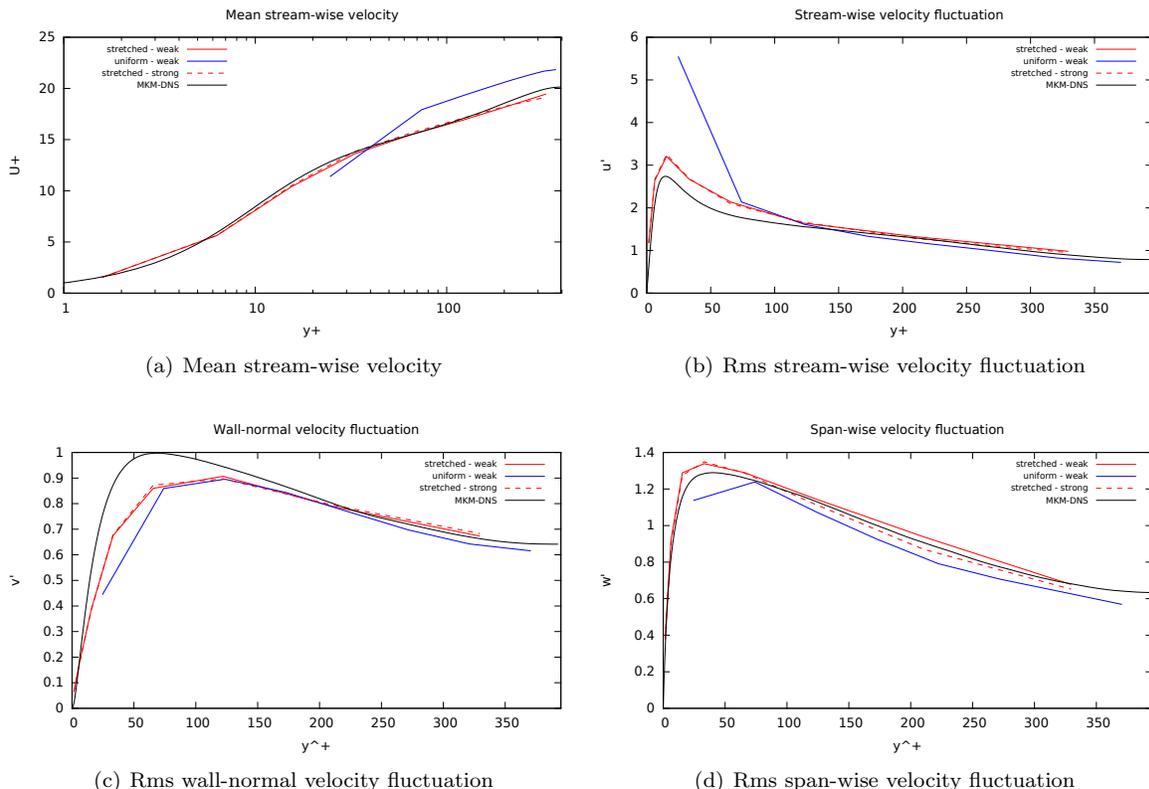


Figure 9: Mean stream-wise velocity and rms velocity fluctuations using different mesh types.

airfoils are widely used as a benchmark of a real life problem, since many airplane and wind turbines wings are based on these profiles. Both numerical simulations and experiments in wind tunnels have been done with several configurations of this type of airfoil. The NACA 0012 is one of the most used configurations (see, e.g., [55, 56, 57], but others like the NACA 4412 are also widely studied (see, e.g., [58, 59]). Another type of airfoil that is also used as a benchmark in the literature is the Aerospatiale-A airfoil, used for example in the LESFOIL project [60].

In this type of problems the mesh design plays an important role. In particular, a proper spacing has to be used at the boundary layer. If a strong definition of the Dirichlet boundary conditions is considered, we say that the simulation is a *Wall-Resolved LES* (WRLES). According to [61], the coherent structures that appear in the turbulent boundary layer can be captured with a WRLES method if the near-wall node of the mesh is located at a wall distance  $y^+ < 2$  and the stream-wise cell size is within the range  $50 < \Delta x^+ < 150$ . Otherwise, when a wall law model is used considering weak imposition of the Dirichlet boundary conditions, we say that the simulation is a *Wall-Modeled LES* (WMLES).

Our concern is to check the feasibility of the proposed method as a WMLES method for the simulation of turbulent flows around an aircraft. In such simulations many flow characteristics are stressed, dealing not simply with turbulent flows, but also with other difficulties, like the presence of very thin laminar boundary layers, transition from laminar to turbulent regimes, wall-bounded flow or flow that separates from curved surfaces. The fact is that LES methods have been shown to be very useful for flows where the turbulent structure is dominated by the large-scale structures, like the TGV test presented in Sect. 5.1 or homogeneous turbulent problems (see [13]). Despite that, the simulation of near-wall flows using LES methods has been demonstrated to be a challenging task, mainly because of the complex physical phenomena, like the reduction of the large scales structures near the wall, the flow anisotropy, or the mesh sensitivity to high aspect ratios.

Many works have been done discussing the suitability of LES methods for the simulation of flow around an airfoil. This is the case of the European project LESFOIL [60], where nine different academic

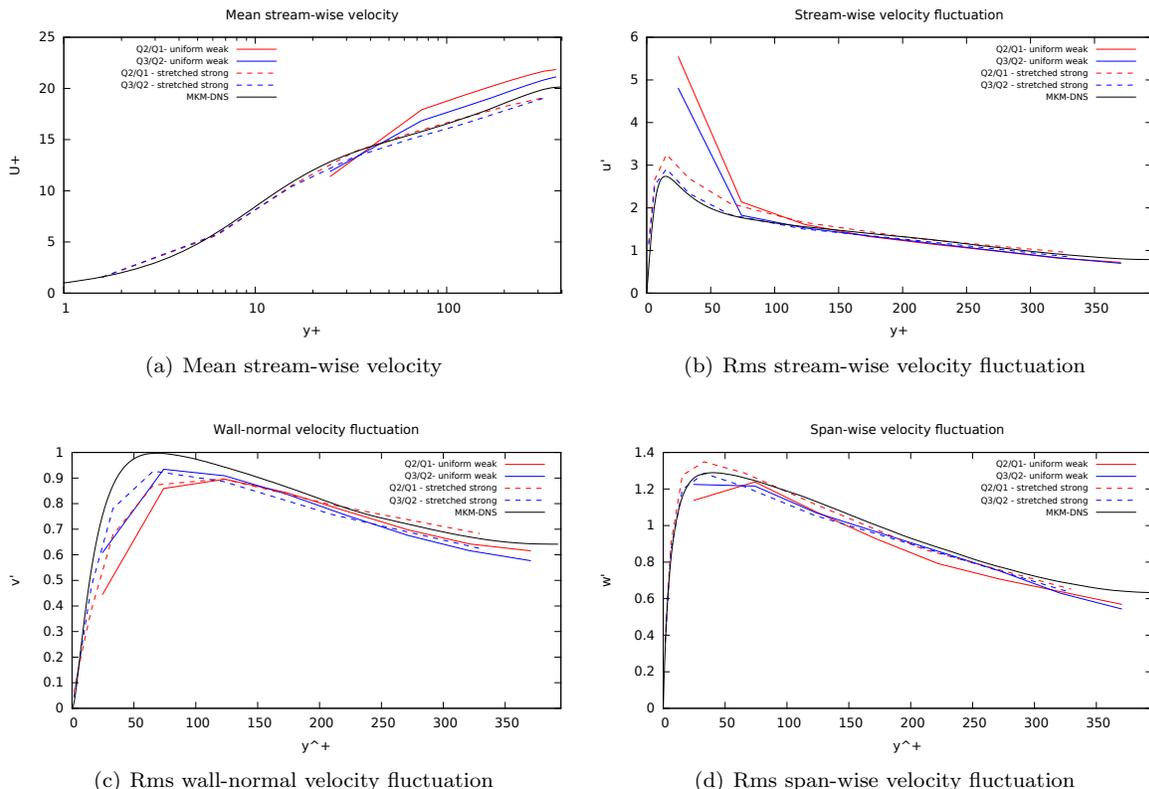


Figure 10: Mean stream-wise velocity and rms velocity fluctuations refining the mesh.

and industrial groups worked with the aim to identify the potential of LES as a prediction method for separation in high-Reynolds-number airfoil flows. In this case, the simulated geometry was the Aerospatiale-A airfoil operating at a chord Reynolds number equal to  $2.1 \cdot 10^6$  with an angle of attack equal to  $13.3^\circ$  and a Mach number of 0.15. Other authors have been worked with NACA type airfoils. This is the case of [62, 63, 64], where the use of LES method for the simulation of the flow around a NACA4412 airfoil is studied using unstructured, structured and semi-structured grids, respectively.

In this paper, the current test has the intention to serve as a proof of concept of the proposed methods for realistic applications. Although most of the works that can be found in the literature deal with challenging configurations, i.e. high Reynolds numbers and high angles of attack (see [62, 63, 64, 65]), we will restrict ourselves to a more modest scenario. That is to consider low Reynolds number with moderate angles of attack. The main reason for considering this case is the possibility to compare the results obtained with a weak treatment of the Dirichlet boundary conditions against the obtained with a strong imposition of the Dirichlet boundaries, without the need of spending too many computational resources. Furthermore, keeping the proof of concep aim in mind, we are not interested on long term computations, which are required to obtain averaged values. Thus, we only compare instantaneous quantities obtained once the turbulence is fully developed.

### 5.3.1. Test setting

We consider the use of a convection-stabilized mixed FE formulation described in Sect. 2.3. Here we also use a weak imposition of the Dirichlet boundary conditions for both the tangential and normal velocity components, detailed in Sect. 2.2. A SRK scheme is used to integrate the equations in time, as described in Sect. 3. In this test we study the flow around a NACA 0012 airfoil with a span-wise length of  $0.16c$ , being  $c$  the chord length. In 11(a) we depict the geometry of the studied airfoil. The computational domain is defined with an inlet and outlet boundaries around 10 chord lengths away from the airfoil surface. At the inlet boundary a free-stream velocity  $U_\infty = 10$  has been settled, while

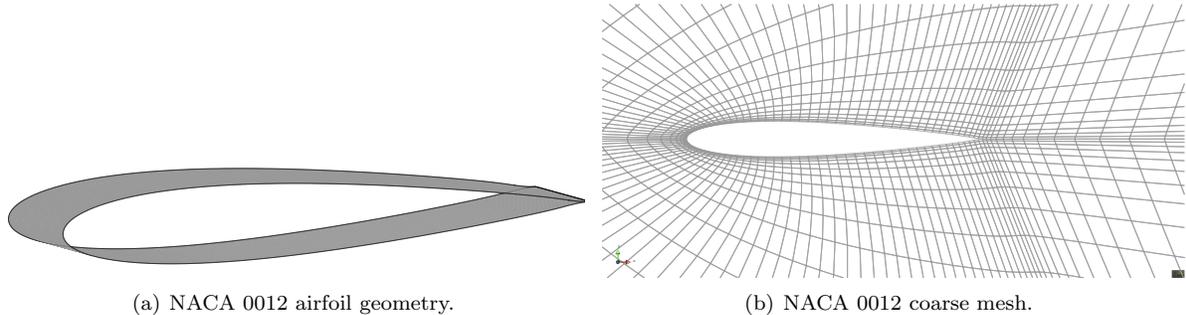


Figure 11: NACA 0012 airfoil geometry and coarse mesh.

the outlet boundary has left free. We consider an angle of attack of  $\alpha = 6$  deg and a Reynolds number based on the chord length  $Re_c = 2.3 \cdot 10^4$ , the same used in [66, 67]. The simulation of this test is done on a structured C-type mesh around the airfoil profile. We consider two different 2-dimensional meshes of quadrilaterals, with different refinement near the wall, which are extruded along the span-wise direction. The goal of using two different meshes is to assess the performance of the weak imposition of the Dirichlet boundary conditions at the airfoil surface when refining the mesh. We define the meshes in such a way that we have fine enough elements around the airfoil surface, the turbulent boundary layer, and the wake region, while coarse elements are used in the far field region. In 11(b) we depict a view of the coarsest mesh. Focusing on the finest mesh, at the leading edge, the near wall-node is located at  $y \sim 2.0 \cdot 10^{-4}c$ , leading to a wall distance of  $y^+ < 1$ . This distance is kept almost constant at the laminar region and it is increased constantly until it reaches the maximum of  $y \sim 2.0 \cdot 10^{-3}$  at the trailing edge, where the wall distance is less restrictive. The maximum stream-wise elemental length is  $\Delta x \sim 0.028c$  located at the suction side of the airfoil, giving a normalized distance of  $\Delta x^+ \sim 40$ . We see that the mesh sizes satisfy the conditions needed to capture the boundary layer phenomena. Regarding the coarsest mesh, the near wall-node is located at  $y \sim 1.2 \cdot 10^{-3}c$  leading to a wall distance of the order of  $y^+ \sim 2$ . This distance is constantly increased until it reaches a value of  $y \sim 6.4 \cdot 10^{-3}c$  at the trailing edge, much greater than the finest mesh. In this mesh, the maximum stream-wise elemental length is  $\Delta x \sim 0.032c$  that is equivalent to a normalized distance of  $\Delta x^+ \sim 50$ . We only compute the 3D test on the coarsest mesh, with a constant span-wise elemental length equal to  $\Delta z \sim 0.02c$ .

The spatial discretization is done using  $Q_2/Q_1$  mixed FEs. Here we use the same algorithmic parameters,  $c_1 = 12.0$ ,  $c_2 = 8.0$  and  $c_c = 32.0$ , according to [10]. The problem is solved using the IMEX version of the SRK method introduced in the previous tests with a  $(\beta-3)$  scheme and an adaptive time stepping technique, as exposed in Sect. 3.4. The adaptive time step size automatically evolves, tending to  $\delta t = 5.0 \cdot 10^{-4}c/U_\infty$ , giving a maximum hyperbolic CFL around 0.3.

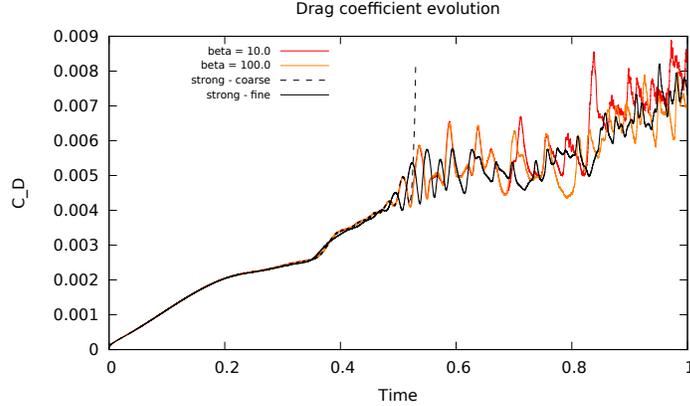
### 5.3.2. Simulation results and discussion

In order to check the suitability of the proposed methods we start with a 2-dimensional simulation comparing the results obtained with the finest and coarsest meshes. For such computations, we begin with an initial solution that has been computed solving the Stokes problem. We solve the problem from  $t = 0.0$  to  $t = 1.0$  which is equivalent to 10 time units (1 time unit =  $c/U_\infty$ ).

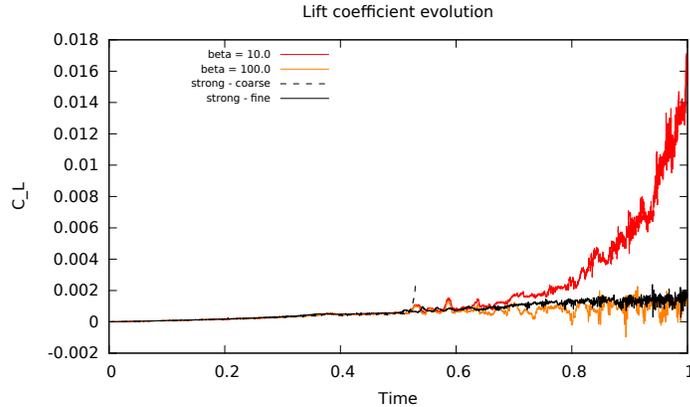
*Effect of  $\beta$  in a 2D mesh with weak boundary conditions.* Once analyzed the effect of the constant  $c_c$  on the solution, we proceed to check the effect of the  $\beta$  constant that appear on the definition of the weak imposition of the wall-normal component on the Dirichlet boundary. In this case we select four different values,  $\beta = 10, 100$  and we solve the problem on the coarse mesh from  $t = 0.0$  to  $t = 1.0$ . The case in which  $\beta = 1000$  have been discarded since the computational cost is much greater than the other two.

In this analysis, as it is seen in 12, very little differences can be observed during the laminar transition to turbulent flow for the different values of  $\beta$ . Furthermore, if we compare against the solution obtained with strong imposition of the boundary conditions using the fine mesh, we see that the results are very similar for the laminar transition. Things change when the flow starts evolving to turbulent behaviour, around  $t = 0.5$ , where we see that the differences between the different parameters selection begin to

increase. The solution with strong boundary conditions in a coarse mesh explodes when the turbulence starts developing; it is expected, since no energy dissipation is produced on the airfoil law. The cases in which we consider a weak boundary imposition are stable, with a drag coefficient similar the one obtained with strong boundary conditions in a finer mesh, see 12(a). However, if we look at 12(b), we see that for  $\beta = 10$  the lift coefficient starts increasing more than the reference solution, while for  $\beta = 100$  this coefficient remains of the same order as in the fine mesh (with strong boundary conditions). Given these results, in the following computations we will use  $\beta = 100$ .



(a) Drag coefficient.



(b) Lift coefficient.

Figure 12: Drag and Lift coefficients for the 2D computation for different  $\beta$  values.

*Instantaneous flow fields for the 3D case.* Once analyzed the effect of using weak Dirichlet boundary conditions on a two-dimensional mesh, we check the performance for the 3D case. Here we use as an initial solution the extruded solution of the 2D case at  $t = 1.0$ . Then, we let the flow evolve until it reaches a 3D description, which occurs after about one time unit. In this section we will analyze the instantaneous flow fields, trying to identify the turbulent structures that are generated along the wing.

In 13 we can see the velocity isosurfaces for  $\|\mathbf{u}\| = 25.0$  colored by the pressure value. In this figure we can clearly see the flow transition from a laminar regime at the leading edge to a turbulent flow, starting the 3D flow development near the maximum wing thickness point.

To study the structures that arise in the turbulent flow, we use the  $Q$ -criterion defined as  $Q = \frac{1}{2}(|\omega|^2 - |\varepsilon(\mathbf{u})|^2)$ , being  $\omega$  the vorticity and  $\varepsilon$  the strain rate tensor. The isosurface of  $Q = 5 \cdot 10^5$  is depicted in 14, where we can see the generation of 2D coherent vortices that break up near the maximum thickness point, turning into hairpin vortices. Similar results are obtained in [66], where a much finer mesh is used to solve the same problem. In 15 we present a top view of the  $Q = 5 \cdot 10^5$  isosurface. There,

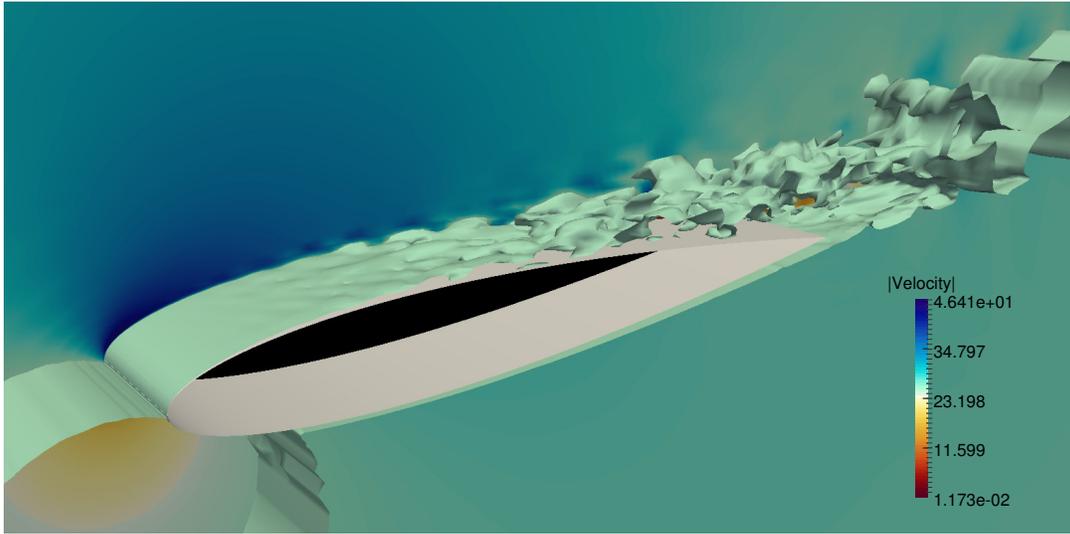


Figure 13: Velocity isosurface for  $\|\mathbf{u}\| = 25.0$ .

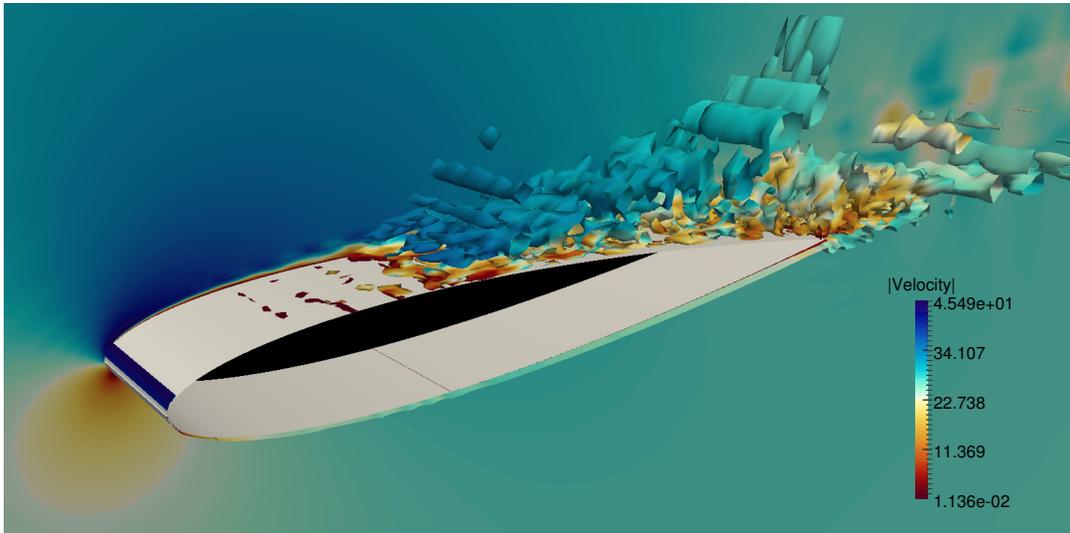


Figure 14: Q-criterion isosurface for  $Q = 5 \cdot 10^5$  and colored by velocity magnitude.

we can see the position of the first 2D coherent vortex, located at  $x/c = 0.37$ . This result is similar to the obtained in [66], where the first 2D vortex is located at  $x/c = 0.41$  on a much finer mesh.

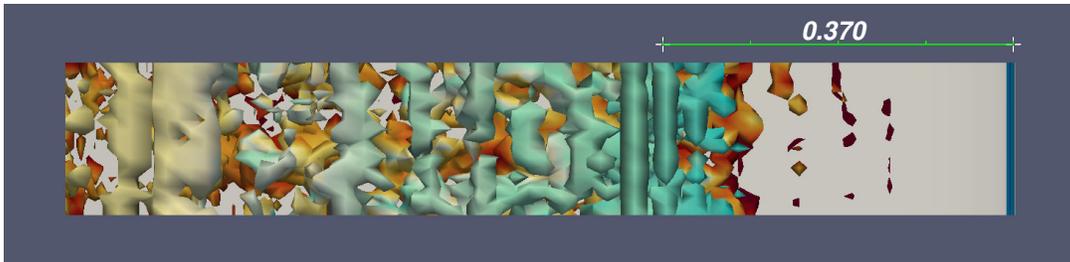


Figure 15: Top view of Q-criterion isosurface for  $Q = 5 \cdot 10^5$ .

## 6. Conclusions

In this work, we have considered convection-stabilized mixed FEs combined with a weak imposition of the boundary conditions that include a wall law model. Convection-stabilized FE methods have been proved to be accurate LES schemes [10]. The motivation to explore the combination with weak boundary conditions is to keep under-resolved LES meshes on boundary layers, in order to maintain a low computational cost compared to fully resolved DNS simulations. On the other hand, SRK time integration schemes have been used for the time integration. SRK methods have recently been proposed in [25]. In this work, we analyze for the first time its behavior for turbulent flow simulations. These schemes are computationally appealing because they segregate velocity and pressure computations at the time integration level, keeping an arbitrary high equal-order accuracy for both unknowns. Finally, block-preconditioning and BDDC preconditioners have been used to end up with highly scalable solvers. The output of this combination is a highly scalable and accurate high-performance framework for the LES simulation of incompressible flows.

We note that the combination of these building blocks has a rationale behind it. SRK schemes can only be applied to index-2 systems, whereas typical pressure-stabilized FE solvers destroy this property. Thus, mixed FE methods are required, whereas convection stabilization is required to produce stable schemes at high Reynolds numbers and model the effect of the subgrid scales. On the other hand, the use of IMEX SRK (with explicit convection) schemes with a block-preconditioner solver for the pressure solver is needed to reduce the linear systems to be solved to coercive problems. It is possible to find scalable preconditioners for these problems, e.g., multilevel domain decomposition solvers, whereas scalable solvers for the indefinite monolithic problem can only be found in particular situations [45].

The application of SRK methods to the simulation of turbulent incompressible flows has been tested with the TGV benchmark, showing that practically the same results are achieved when considering a SRK time integration versus a monolithic scheme. The convergence of the proposed LES schemes to the DNS solution has also been demonstrated when a refinement both in  $h$  and  $p$  is considered, giving more accurate results the  $p$ -refinement (as expected for smooth enough solutions). One of the advantages of using SRK schemes is the easy implementation of time adaptive techniques that allow the automatic time step adaptation to the numerical and physical requirements. This issue has also been addressed for the TGV test, where we have seen that the method is able to adapt to the numerical restriction given by the CFL conditions and to the physical requirements given by the change in the solution. The combination of SRK schemes, block-preconditioning, and BDDC preconditioners lead to excellent weak scalability results, tested for TGV benchmark up to 8000 cores using uniform meshes. Furthermore, a three-level MLBDDC algorithm could be used to achieve weak scalability on a higher number of processors, following the ideas in [28].

We have also checked the performance of the proposed schemes for the wall-bounded TCF test. In this case, we have included the use of weak Dirichlet boundary conditions with a wall law based tangential traction and a weak imposition of the wall-normal component. This approach leads to acceptable results, even when wall-unresolved uniform meshes are considered. Further, on stretched meshes, the use of strong or weak Dirichlet boundary conditions lead to very similar results.

We have seen that the weak imposition of the Dirichlet boundary conditions is suitable when SRK methods are considered. First, because the continuity equation is not polluted by the strong imposition of the temporal derivative of the velocity divergence on the boundary. This occurs when time-dependent boundary conditions are imposed on the Dirichlet boundary. Furthermore, the weak imposition of the velocity wall-normal component considered in this work, avoids the complexity of imposing strongly such component on curved boundaries.

## Acknowledgements

This work has been partially funded by the European Research Council under the FP7 Programme Ideas through the Starting Grant No. 258443 - COMFUS: Computational Methods for Fusion Technology. O. Colomés gratefully acknowledges the support received from the Catalan Government through a FI fellowship.

## References

- [1] P. Sagaut, *Large Eddy Simulations for Incompressible Flows*, Vol. 3, Springer Berlin, 2000.
- [2] J. P. Boris, F. F. Grinstein, E. S. Oran, R. L. Kolbe, New insights into large eddy simulation, *Fluid Dynamics Research* 10 (4-6) (1992) 199.
- [3] T. J. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, *Computer Methods in Applied Mechanics and Engineering* 127 (1-4) (1995) 387–401.
- [4] T. J. Hughes, G. R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method - a paradigm for computational mechanics, *Computer Methods in Applied Mechanics and Engineering* 166 (1–2) (1998) 3–24.
- [5] T. J. R. Hughes, L. Mazzei, A. A. Oberai, A. A. Wray, The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence, *Physics of Fluids* 13 (2) (2001) 505–512.
- [6] T. J. R. Hughes, A. A. Oberai, L. Mazzei, Large eddy simulation of turbulent channel flows by the variational multiscale method, *Physics of Fluids* 13 (6) (2001) 1784–1799.
- [7] R. Codina, Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, *Computer Methods in Applied Mechanics and Engineering* 191 (39–40) (2002) 4295–4321.
- [8] R. Codina, Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods, *Computer Methods in Applied Mechanics and Engineering* 190 (13–14) (2000) 1579–1599.
- [9] R. Codina, Analysis of a stabilized finite element approximation of the Oseen equations using orthogonal subscales, *Applied Numerical Mathematics* 58 (3) (2008) 264–283.
- [10] O. Colomés, S. Badia, J. Principe, Mixed finite element methods with convection stabilization for the large eddy simulation of incompressible turbulent flows, *Computer Methods in Applied Mechanics and Engineering* 304 (2016) 294 – 318.
- [11] S. Badia, On stabilized finite element methods based on the Scott–Zhang projector. Circumventing the inf–sup condition for the Stokes problem, *Computer Methods in Applied Mechanics and Engineering* 247–248 (0) (2012) 65–72.
- [12] M. Braack, E. Burman, Local Projection Stabilization for the Oseen Problem and its Interpretation as a Variational Multiscale Method, *SIAM Journal on Numerical Analysis* 43 (6) (2006) 2544–2566.
- [13] O. Colomés, S. Badia, R. Codina, J. Principe, Assessment of variational multiscale models for the large eddy simulation of turbulent incompressible flows, *Computer Methods in Applied Mechanics and Engineering* 285 (2015) 32–63.
- [14] Y. Bazilevs, T. J. R. Hughes, Weak imposition of Dirichlet boundary conditions in fluid mechanics, *Computers & Fluids* 36 (1) (2007) 12–26.
- [15] Y. Bazilevs, C. Michler, V. M. Calo, T. J. R. Hughes, Weak Dirichlet boundary conditions for wall-bounded turbulent flows, *Computer Methods in Applied Mechanics and Engineering* 196 (49–52) (2007) 4853–4862.
- [16] V. John, Slip with friction and penetration with resistance boundary conditions for the Navier–Stokes equations—numerical tests and aspects of the implementation, *Journal of Computational and Applied Mathematics* 147 (2) (2002) 287–300.
- [17] L. Davidson, LESFOIL: A European Project on Large Eddy Simulations Around a High-lift Airfoil at High Reynolds Number, in: *Chalmers Publication Library (CPL)*, 2000.

- [18] Y. Bazilevs, C. Michler, V. Calo, T. Hughes, Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes, *Computer Methods in Applied Mechanics and Engineering* 199 (13) (2010) 780–790.
- [19] M.-C. Hsu, I. Akkerman, Y. Bazilevs, Wind turbine aerodynamics using ale–vms: validation and the role of weakly enforced boundary conditions, *Computational Mechanics* 50 (4) (2012) 499–511.
- [20] H. Choi, P. Moin, Effects of the computational time step on numerical solutions of turbulent flow, *Journal of Computational Physics* 113 (1) (1994) 1–4.
- [21] H. Le, P. Moin, An improvement of fractional step methods for the incompressible Navier-Stokes equations, *Journal of Computational Physics* 92 (2) (1991) 369–379.
- [22] D. Forti, L. Dedè, Semi-implicit BDF time discretization of the Navier–Stokes equations with VMS-LES modeling in a High Performance Computing framework, *Computers & Fluids* 117 (2015) 168–182.
- [23] S. Badia, R. Codina, Algebraic Pressure Segregation Methods for the Incompressible Navier-Stokes Equations, *Archives of Computational Methods in Engineering* 15 (3) (2008) 343–369.
- [24] S. Badia, A. F. Martín, R. Planas, Block recursive LU preconditioners for the thermally coupled incompressible inductionless MHD problem, *Journal of Computational Physics* 274 (2014) 562–591.
- [25] O. Colomé, S. Badia, Segregated Runge–Kutta methods for the incompressible Navier–Stokes equations, *International Journal for Numerical Methods in Engineering* 105 (5) (2016) 372–400.
- [26] C. R. Dohrmann, A Preconditioner for Substructuring Based on Constrained Energy Minimization, *SIAM Journal on Scientific Computing* 25 (1) (2003) 246–258.
- [27] S. Badia, A. F. Martín, J. Principe, A Highly Scalable Parallel Implementation of Balancing Domain Decomposition by Constraints, *SIAM Journal on Scientific Computing* 36 (2) (2014) 190–218.
- [28] S. Badia, A. F. Martín, J. Principe, Multilevel balancing domain decomposition at extreme scales, *SIAM Journal on Scientific Computing* 38 (1) (2016) 22–52.
- [29] P. R. Spalart, Strategies for turbulence modelling and simulations, *International Journal of Heat and Fluid Flow* 21 (3) (2000) 252–263.
- [30] P. Spalart, W. Jou, M. Strelets, S. Allmaras, Comments on the feasibility of les for wings, and on a hybrid rans/les approach, *Advances in DNS/LES* 1 (1997) 4–8.
- [31] S. B. Pope, *Turbulent Flows*, Cambridge University Press, 2000.
- [32] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 36 (1) (1971) 9–15.
- [33] R. Stenberg, On some techniques for approximating boundary conditions in the finite element method, *Journal of Computational and Applied Mathematics* 63 (1–3) (1995) 139–148.
- [34] D. Arnold, An Interior Penalty Finite Element Method with Discontinuous Elements, *SIAM Journal on Numerical Analysis* 19 (4) (1982) 742–760.
- [35] A. Linke, Collision in a cross-shaped domain - A steady 2d Navier-Stokes example demonstrating the importance of mass conservation in CFD, *Computer Methods in Applied Mechanics and Engineering* 198 (41-44) (2009) 3278–3286.
- [36] M. Olshanskii, A. Reusken, Grad-div stabilization for Stokes equations, *Mathematics of Computation* 73 (248) (2004) 1699–1718.

- [37] T. Heister, G. Rapin, Efficient augmented Lagrangian-type preconditioning for the Oseen problem using Grad-Div stabilization, *International Journal for Numerical Methods in Fluids* 71 (1) (2013) 118–134.
- [38] G. Söderlind, Automatic Control and Adaptive Time-Stepping, *Numerical Algorithms* 31 (1-4) (2002) 281–310.
- [39] K. Gustafsson, Control theoretic techniques for stepsize selection in explicit Runge-Kutta methods, *ACM Trans. Math. Softw.* 17 (4) (1991) 533–554.
- [40] K. Gustafsson, Control-theoretic Techniques for Stepsize Selection in Implicit Runge-Kutta Methods, *ACM Trans. Math. Softw.* 20 (4) (1994) 496–517.
- [41] H. C. Elman, D. J. Silvester, A. J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Oxford University Press, 2005.
- [42] S. C. Brenner, R. Scott, *The Mathematical Theory of Finite Element Methods*, softcover reprint of hardcover 3rd ed. 2008 Edition, Springer, 2010.
- [43] J. Mandel, B. Sousedik, C. Dohrmann, Multispace and multilevel BDDC, *Computing* 83 (2) (2008) 55–85.
- [44] S. Badia, H. Nguyen, Balancing Domain Decomposition by perturbation, Submitted.
- [45] J. Li, O. Widlund, BDDC Algorithms for Incompressible Stokes Equations, *SIAM Journal on Numerical Analysis* 44 (6) (2006) 2432–2455.
- [46] S. Badia, M. Olm, Space-time balancing domain decomposition, Submitted.
- [47] S. Badia, A. F. Martín, J. Principe, On the scalability of inexact balancing domain decomposition by constraints with overlapped coarse/fine corrections, *Parallel Computing* 50 (2015) 1–24.
- [48] M. Brachet, Direct simulation of three-dimensional turbulence in the Taylor-Green vortex, *Fluid Dynamics Research* 8 (1–4) (1991) 1–8.
- [49] S. Badia, A. F. Martín, J. Principe, Implementation and Scalability Analysis of Balancing Domain Decomposition Methods, *Archives of Computational Methods in Engineering* 20 (3) (2013) 239–262.
- [50] R. D. Moser, J. Kim, N. N. Mansour, Direct numerical simulation of turbulent channel flow up to  $Re_\tau=590$ , *Physics of Fluids* 11 (4) (1999) 943–945.
- [51] J. Kim, P. Moin, R. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, *Journal of Fluid Mechanics* 177 (1987) 133–166.
- [52] P. Moin, J. Kim, On the numerical solution of time-dependent viscous incompressible fluid flows involving solid boundaries, *Journal of Computational Physics* 35 (3) (1980) 381–392.
- [53] V. Gravemeier, M. W. Gee, M. Kronbichler, W. A. Wall, An algebraic variational multiscale-multigrid method for large eddy simulation of turbulent flow, *Computer Methods in Applied Mechanics and Engineering* 199 (13–16) (2010) 853–864.
- [54] E. N. Jacobs, K. E. Ward, R. M. Pinkerton, The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel, naca-report-460.
- [55] R. E. Sheldahl, P. C. Klimas, Aerodynamic Characteristics of Seven Symmetrical Airfoil Sections Through 180-Degree Angle of Attack for Use in Aerodynamic Analysis of Vertical Axis Wind Turbines, Tech. Rep. SAND-80-2114, Sandia National Labs., Albuquerque, NM (USA) (1981).
- [56] W. McCroskey, A critical assessment of wind tunnel results for the NACA 0012 airfoil, Tech. rep., DTIC Document (1987).

- [57] J. A. Rivera, B. E. Dansberry, R. M. Bennett, M. H. Durham, W. A. Silva, NACA0012 benchmark model experimental flutter results with unsteady pressure distributions, National Aeronautics and Space Administration, Langley Research Center, 1992.
- [58] A. J. Wadcock, Investigation of low-speed turbulent separated flow around airfoils, Tech. rep. (1987).
- [59] R. C. Hastings, B. R. Williams, Studies of the Flow Field near a NACA 4412 Aerofoil at Nearly Maximum Lift., Tech. rep. (1984).
- [60] L. Davidson, D. Cokljat, J. Fröhlich, M. A. Leschziner, C. Mellen, W. Rodi, LESFOIL: Large Eddy Simulation of Flow Around a High Lift Airfoil: Results of the Project LESFOIL Supported by the European Union 1998 – 2001, Springer Science & Business Media, 2012.
- [61] U. Piomelli, J. R. Chasnov, Large-Eddy Simulations: Theory and Applications, in: M. Hallböck, D. S. Henningson, A. V. Johansson, P. H. Alfredsson (Eds.), Turbulence and Transition Modelling, no. 2 in ERCOFTAC Series, Springer Netherlands, 1996, pp. 269–336.
- [62] K. E. Jansen, A stabilized finite element method for computing turbulence, Computer Methods in Applied Mechanics and Engineering 174 (3–4) (1999) 299–317.
- [63] H.-J. Kaltenbach, H. Choi, Large-eddy simulation of flow around an airfoil on a structured mesh, Tech. rep. (1995).
- [64] S. Schmidt, M. Franke, F. Thiele, Assessment of SGS models in LES applied to a NACA 4412 airfoil, in: 39th Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics.
- [65] C. P. Mellen, J. Fröhlich, W. Rodi, Lessons from LESFOIL project on large-eddy simulation of flow around an airfoil, AIAA journal 41 (4) (2003) 573–581.
- [66] R. Kojima, T. Nonomura, A. Oyama, K. Fujii, Large-Eddy Simulation of Low-Reynolds-Number Flow Over Thick and Thin NACA Airfoils, Journal of Aircraft 50 (1) (2013) 187–196.
- [67] D.-H. Kim, J.-H. Yang, Boundary Layer and Near-Wake Measurements of NACA 0012 Airfoil at Low Reynolds Numbers (AIAA) (2009).