Mixed finite element methods with convection stabilization for the large eddy simulation of incompressible turbulent flows

Oriol Colomés[a], Santiago Badia[a,b], Javier Principe[a,b]

[a]*Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Parc Mediterrani de la Tecnologia, UPC, Esteve Terradas 5, 08860 Castelldefels, Spain ({ocolomes,sbadia,principe}@cimne.upc.edu).*
[b]*Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.*

Abstract

The variational multiscale method thought as an implicit large eddy simulation model for turbulent flows has been shown to be an alternative to the widely used physical-based models. This method is traditionally combined with equal-order velocity-pressure pairs, since it provides pressure stabilization. In this work, we consider a different approach, based on inf-sup stable elements and convection-only stabilization. In order to do so, we consider a symmetric projection stabilization of the convective term using a orthogonal subscale decomposition. The accuracy and efficiency of this method compared with residual-based algebraic subgrid scales and orthogonal subscales methods for equal-order interpolation is assessed in this paper. Moreover, when inf-sup stable elements are used, the grad-div stabilization term has been shown to be essential to guarantee accurate solutions. Hence, a study of the influence of such term in the large eddy simulation of turbulent incompressible flows is also performed. Furthermore, a recursive block preconditioning strategy has been considered for the resolution of the problem with an implicit treatment of the projection terms. Two different benchmark tests have been solved: the Taylor-Green Vortex flow with $Re = 1600$, and the Turbulent Channel Flow at $Re_\tau = 395$ and $Re_\tau = 590$.

Keywords: Large eddy simulation, turbulence, variational multiscale, block recursive preconditioning, grad-div stabilization

1. Introduction

The simulation of incompressible turbulent flows is needed to solve many problems in science and engineering, e.g., flows around aircrafts, wind turbines, cars, bridges, etc. The computational fluid dynamics field is constantly evolving and searching the best way to achieve faster and cheaper flow simulations, taking advantage of the increasing computational power of supercomputers.

Turbulent flows are characterized by having multiple scales, both in space and time, that make their simulation a really challenging task. Direct Numerical Simulations (DNS) of turbulent flows are used to capture the physical phenomena at all scales, even the smallest ones, but they consume a large amount of computational resources. A technique that is increasingly used and that saves a lot of computational cost is the Large Eddy Simulation (LES), which basically consists in separating the flow in a coarse scale and a fine one, simulating the coarser and modelling the finer [1]. The model of the fine scales in a LES method can be developed by a physical approach, which is defined taking into account the physical phenomena that take place on the smallest scales, or by a purely numerical approach that does not introduce any modification to the governing equations at the continuous level. This last numerical approach is commonly denoted as Implicit LES (ILES), see for instance [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. In the context of the incompressible Navier-Stokes equations there are two well known instabilities. First, there is the need to satisfy a compatibility between velocity and pressure approximation spaces, i.e., the *inf-sup* condition, which is not satisfied when equal order interpolation is used. Second, we have the instabilities that appear when convective dominated flows are simulated, which can be solved by adding some kind of extra dissipation to the problem, as in, e.g., [5]. There is also a much less known loss of accuracy in

the discrete velocity due to poor mass conservation [6] which can be linked to a poor resolution of the pressure typical of mixed interpolations that satisfy the inf-sup condition [7].

The VMS method was firstly thought as an ILES method in [8, 9, 10] and, since then, several VMS methods have been developed and used as ILES. We can distinguish between those that introduce a three scale decomposition into resolved large and small scales and unresolved scales [11, 12, 13, 14], with a Smagorinsky type model for the influence of unresolved scales onto the small resolved ones, and those that introduce a two scale decomposition into resolved and unresolved ones [15, 16] using a residual based or projection based model of the unresolved scales to account for their influence into the resolved ones.

In [17], a two scale VMS approach through Orthogonal SubScales (OSS) was firstly introduced. The main idea of the OSS method is to select the space of small scales orthogonal to the FE space, in contrast to the traditional choice of taking the subscales proportional to the residual, which is called Algebraic SubGrid Scale (ASGS) in [17]. Apart from the choice of the space of subscales, their time dependency and the VMS splitting of nonlinear terms was studied in [18]. Several combinations of these modelling possibilities where exhaustively assessed for homogeneous and wall bounded turbulent flows in [16]. In that work, an explicit algorithm to compute the orthogonal projections was used and the projection of the whole residual was considered.

An alternative definition of the OSS method was proposed in [19] using a term-by-term stabilization that does not involve the full residual. A similar term-by-term stabilization approach was followed in [20], where the Local Projection Stabilization (LPS) method was introduced. This type of techniques are also known as symmetric projection stabilization, and the key ingredient that leads to different schemes is the definition of such projection [20, 21]. One of the main interests of a term-by-term stabilization is that one can avoid the addition of the pressure gradient stabilization term when using *inf-sup* stable (ISS) velocity-pressure pairs. Another advantage of the *term-by-term* stabilization methods is that the projection can be easily treated as implicit, without having all the residual terms coupled. ISS pairs with SUPG stabilization of the convection were considered in [22], where the term *grad-div* was first used. A comparison of the use of equal-order or ISS pairs with LPS was assessed for laminar flows in [23], concluding that the grad-div stabilization term is more relevant than the convective stabilization term when ISS FEs are used. The same conclusion was pointed out in [13], where the turbulent channel flow is studied using a projection-based and a bubble-based FE VMS method introducing a Smagorinsky type model to stabilize convection.

The main goal of this paper is to assess for the first time the accuracy and efficiency of convection-stabilized ISS schemes as ILES methods, where symmetric projection stabilization is used. In particular, we analyze the term-by-term OSS method with implicit treatment of the projection for turbulent flows. We also analyze the influence of the grad-div stabilization on the accuracy of the method. For ISS discretizations, the influence of this term on the mass conservation is well known [6] but it also influences the computational cost of the linear solvers [24, 25]. In this respect, we present a block preconditioning strategy that makes use of recursive block factorizations [26] to deal with the implicit projections and with the saddle point structure of the velocity-pressure coupling, which can also be applied to equal order interpolation with pressure stabilization. The comparison of the results is made with respect to those obtained using the residual-based ASGS for which we also use a block preconditioning strategy.

This work is organized as follows. The Navier-Stokes equations together with some notation used in the paper are stated in Sect. 2. The VMS framework is introduced in Sect. 3, which includes the final discrete formulation of the ASGS method, given in Sect. 3.1, the term-by-term OSS given in Sect. 3.2, the term-by-term OSS with ISS elements given in Sect. 3.3 and also a brief discussion of known properties of the grad-div stabilization given in Sect. 3.4. The recursive block iterative strategy proposed to solve the linear system of the monolithic problem is presented in Sect. 4. The numerical results are shown in Sect. 5, where two different turbulent tests are analyzed: the Taylor-Green Vortex flow in Sect. 5.2 and the Turbulent Channel Flow, both for a $Re_\tau = 395$ in Sect. 5.3, and for a $Re_\tau = 590$ in Sect. 5.4. Finally, some conclusions are pointed out in Sect. 6.

2. Problem statement

Let Ω 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 steady Navier-Stokes problem consists in finding the velocity field $\mathbf{u}$ and the pressure field p such that

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

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

with $\mathbf{f}$ the force vector and ν 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 Γ is divided into the Dirichlet (Γ_D) and the Neumann (Γ_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

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

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

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

$\mathbf{n}$ being the unit outward vector normal to Γ and $\mathbf{u}_0(x)$ satisfying $\nabla \cdot \mathbf{u}_0 = 0$.

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

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

and induced norm $\|u\|_{L^2(\Omega)} \equiv \|u\| = (u, u)^{1/2}$. 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 Γ_D and $H_g^1(\Omega)$ is the set of functions in $H^1(\Omega)$ whose traces satisfy (3). 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 space of time-dependent functions such that their X-norm is in $L^p(0, T)$. Bold letters are used for spaces of vector functions.

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

$$(\partial_t\mathbf{u}, \mathbf{v}) + B(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) = \langle \mathbf{f}, \mathbf{v} \rangle \qquad \forall\mathbf{v} \in \mathcal{V}_0, \quad \forall q \in \mathcal{Q}_0, \quad \forall t \in [0, T], \tag{7}$$

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

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

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

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

We refer to [16] for an evaluation of the effect of the skew-symmetry form of the convective term on the results when simulating turbulent incompressible flows.

3. VMS methods

Let us consider a FE partition $\mathcal{T}_h$ of the domain Ω from which we can construct conforming finite dimensional spaces for the velocity $\mathcal{V}_{0,h} \subset \mathcal{V}_0$, $\mathcal{V}_{g,h} \subset \mathcal{V}_g$, and for the pressure $\mathcal{Q}_{0,h} \subset \mathcal{Q}_0$. The Galerkin

FE approximation of the problem (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

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + B(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = \langle \mathbf{f}, \mathbf{v}_h \rangle \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}, \forall q_h \in \mathcal{Q}_{0,h}. \tag{10}$$

In order to overcome the numerical instabilities and, eventually, to bypass the *inf-sup* condition that arises when problem (10) is solved, we use the VMS approach [3, 4] which consists in a two-scale decomposition of spaces $\mathcal{V}_0$, $\mathcal{V}_g$ and $\mathcal{Q}_0$ as $\mathcal{V}_0 = \mathcal{V}_{0,h} \oplus \widetilde{\mathcal{V}}_0$, $\mathcal{V}_g = \mathcal{V}_{g,h} \oplus \widetilde{\mathcal{V}}_g$ and $\mathcal{Q} = \mathcal{Q}_{0,h} \oplus \widetilde{\mathcal{Q}}_0$, where $\widetilde{\mathcal{V}}_0$, $\widetilde{\mathcal{V}}_g$ and $\widetilde{\mathcal{Q}}_0$ are infinite-dimensional spaces that complete the FE spaces in $\mathcal{V}_0$, $\mathcal{V}_g$ and $\mathcal{Q}_0$, respectively. Hereinafter the subscript $(\cdot)_h$ will denote the FE component and the tilde $\widetilde{(\cdot)}$ the subgrid component. Applying the two-scale decomposition to (7) we obtain

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + (\partial_t \tilde{\mathbf{u}}, \mathbf{v}_h) + B(\mathbf{a}; [\mathbf{u}_h, p_h], [\mathbf{v}_h, q_h]) + B(\mathbf{a}; [\tilde{\mathbf{u}}, \tilde{p}], [\mathbf{v}_h, q_h]) = \langle \mathbf{f}, \mathbf{v}_h \rangle, \tag{11}$$

$$(\partial_t \mathbf{u}_h, \tilde{\mathbf{v}}) + (\partial_t \tilde{\mathbf{u}}, \tilde{\mathbf{v}}) + B(\mathbf{a}; [\mathbf{u}_h, p_h], [\tilde{\mathbf{v}}, \tilde{q}]) + B(\mathbf{a}; [\tilde{\mathbf{u}}, \tilde{p}], [\tilde{\mathbf{v}}, \tilde{q}]) = \langle \mathbf{f}, \tilde{\mathbf{v}} \rangle. \tag{12}$$

The strict application of the scale splitting implies $\mathbf{a} = \mathbf{u}_h + \tilde{\mathbf{u}}$, which is referred to as nonlinear subscales in [18], although the standard approach (linear subscales in [18]) is to take $\mathbf{a} = \mathbf{u}_h$ (see also [16]). The fourth term on the left hand side of (11) is actually a sum of four terms that depend on subscales gradients (see (8)). To avoid computing these gradients, these terms are integrated by parts elementwise. Using the notation $(\cdot, \cdot)_h = \sum_{K \in \mathcal{T}_h} (\cdot, \cdot)_K$ and $(\cdot, \cdot)_{\partial h} = \sum_{K \in \mathcal{T}_h} (\cdot, \cdot)_{\partial K}$ and neglecting the boundary terms $(\mathbf{v}_h, \nu \mathbf{n} \cdot \nabla \tilde{\mathbf{u}})_{\partial h}$ and $(q_h, \mathbf{n} \cdot \tilde{\mathbf{u}})_{\partial h}$, we get:

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + (\partial_t \tilde{\mathbf{u}}, \mathbf{v}_h) + B(\mathbf{a}; [\mathbf{u}_h, p_h], [\mathbf{v}_h, q_h]) - (\tilde{\mathbf{u}}, \nu \Delta \mathbf{v}_h + \mathbf{a} \cdot \nabla \mathbf{v}_h + \nabla q_h)_h - (\tilde{p}, \nabla \cdot \mathbf{v}_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle. \tag{13}$$

To obtain (13), the approximation $b(\mathbf{a}, \tilde{\mathbf{u}}, \mathbf{u}_h) \approx -(\tilde{\mathbf{u}}, \mathbf{a} \cdot \nabla \mathbf{v}_h)$ has also been used, which implies neglecting $(\mathbf{v}_h, \mathbf{n} \cdot \mathbf{a}\tilde{\mathbf{u}})_{\partial h}$ and $(\tilde{\mathbf{u}}, \nabla \cdot \mathbf{a} \, \mathbf{v}_h)$. These approximations are discussed in [18].

Problem (13) depends on $\tilde{\mathbf{u}} \in \widetilde{\mathcal{V}}_0$ and on $\tilde{p} \in \widetilde{\mathcal{Q}}_0$, $\widetilde{\mathcal{V}}_0$ and $\widetilde{\mathcal{Q}}_0$ being infinite-dimensional. Therefore, (12) cannot be directly solved, but some modelling steps are needed to obtain a feasible method. In this work we consider the velocity subscale as linear ($\mathbf{a} = \mathbf{u}_h$) and quasi-static, while the pressure subscale is neglected when equal order interpolation is used. Different approaches could be used for the definition of the subscales, see [16] for a deep explanation of the different choices and their numerical evaluation with equal order approximation. Approximating the Navier-Stokes operators by the stabilization parameters τ_m^{-1} and τ_c^{-1} (see for example [18]), the fine scale problem can be written as

$$\tau_m^{-1} \tilde{\mathbf{u}} = \mathcal{P}_{\widetilde{\mathcal{V}}}(\mathbf{R}_u), \tag{14}$$

$$\tau_c^{-1} \tilde{p} = \mathcal{P}_{\widetilde{\mathcal{Q}}}(R_p). \tag{15}$$

In (14)-(15) $\mathcal{P}_*$ denotes the projection onto the space of subscales, which is discussed below. In turn, the vector $\mathbf{R}_u$ is the residual of the Navier-Stokes momentum equation (1) and R_p the residual of the incompressibility constraint (2), defined as

$$\mathbf{R}_u = \mathbf{f} - \partial_t \mathbf{u}_h + \nu \Delta \mathbf{u}_h - \mathbf{u}_h \cdot \nabla \mathbf{u}_h - \nabla p_h, \tag{16}$$

$$R_p = -\nabla \cdot \mathbf{u}_h. \tag{17}$$

Finally, the expressions of the stabilization parameters τ_m and τ_c are

$$\tau_m = \left(\frac{c_1 \nu}{h^2} + \frac{c_2 |\mathbf{u}_h|}{h} \right)^{-1}, \tag{18}$$

$$\tau_c = \frac{h^2}{c_1 \tau_m}. \tag{19}$$

where h is the mesh size and c_1 and c_2 are algorithmic constants. The choice of the value for c_1 and c_2 make a significant difference on the VMS method, specially for turbulent flows, where the energy dissipation introduced by the stabilization method takes an important role in the final solution. In this work, we select $c_1 = 12$ and $c_2 = 2$, following [16].

4

Alternatively to (19), we can reformulate the τ_c definition by

$$\tau_c = c_c \left(\nu + \frac{c_2}{c_1} h |\mathbf{u}_h| \right), \tag{20}$$

which is equivalent to (19) when $c_c = 1.0$.

Introducing $\tilde{\mathbf{u}}$ and $\tilde{p}$ into (13) we get the final discrete problem

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + B_h(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = L_h(\mathbf{v}_h, q_h) \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}, \forall q_h \in \mathcal{Q}_{0,h}, \tag{21}$$

The bilinear form B_h and the linear form L_h depend on the particular VMS method as discussed below. Several results on the numerical analysis of VMS methods are available [27].

3.1. Residual-based ASGS

The space for the subscales $\widetilde{\mathcal{V}}_0$ is determined by the definition of the projection $\mathcal{P}$ appearing in the right-hand side of (14)-(15). The ASGS method is obtained taking the subscales in the space of the residuals, that is, $\mathcal{P}_{\widetilde{\mathcal{V}}} := \mathbf{I}$ and $\mathcal{P}_{\widetilde{\mathcal{Q}}} := I$. The final discrete problem is given by (21) with $B_h = B_{asgs}$ and $L_h = L_{asgs}$ given by

$$B_{asgs}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) := B(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) \tag{22}$$
$$+ (\tau_m(\partial_t \mathbf{u}_h - \nu \Delta \mathbf{u}_h + \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla p_h), \nu \Delta \mathbf{v}_h + \mathbf{u}_h \cdot \nabla \mathbf{v}_h + \nabla q_h),$$
$$L_{asgs}(\mathbf{v}_h, q_h) := \langle \mathbf{f}, \mathbf{v}_h \rangle + (\tau_m \mathbf{f}, \nu \Delta \mathbf{v}_h + \mathbf{u}_h \cdot \nabla \mathbf{v}_h + \nabla q_h). \tag{23}$$

Note that the pressure subscale term has been neglected in (22), that is we have taken $\tau_c = 0$. We have observed in [16] (using equal order interpolation) that this term introduces extra dissipation and does not in general improve the solution significantly.

3.2. Term by term OSS

Another possibility introduced in [17] is to consider the space of the subscales orthogonal to the FE space. This method is mainly motivated by the fact that a stability estimate for the projection onto the FE space of the pressure and the convective terms can already be obtained in the standard Galerkin method and therefore the only "missing" part is the orthogonal one. The Orthogonal Subscales (OSS) method is then obtained taking $\mathcal{P}_{\widetilde{\mathcal{V}}} := \Pi_h^\perp = \mathbf{I} - \Pi_h$ where Π_h is a projection onto the FE space. The L^2 orthogonality between the FE and subscale spaces is guaranteed considering the τ_m-weighted projection

$$(\tau_m \Pi_h(\mathbf{w}), \mathbf{v}_h) = (\tau_m \mathbf{w}, \mathbf{v}_h) \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}. \tag{24}$$

which requires the solution of a linear system defined by a τ_m-weighted mass matrix. Note that for this choice, the residual of the momentum equation does not depend on $\partial_t \mathbf{u}_h$. Likewise, $\mathcal{P}(\mathbf{f})$ in this case is only well defined for $\mathbf{f} \in L^2(\Omega)^d$. In the case of minimum regularity, $\mathbf{f} \in H^{-1}(\Omega)^d$, this term can be simply neglected without upsetting the accuracy of the method.

Remark 3.1. *Alternatively we can consider the standard L^2 projection*

$$(\Pi_h(\mathbf{w}), \mathbf{v}_h) = (\mathbf{w}, \mathbf{v}_h) \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}, \tag{25}$$

modifying the model of the subscales as

$$\tilde{\mathbf{u}} = \mathcal{P}_{\widetilde{\mathcal{V}}}(\tau_m \mathbf{R}_u), \tag{26}$$
$$\tilde{p} = \mathcal{P}_{\widetilde{\mathcal{Q}}}(\tau_c R_p). \tag{27}$$

With this modification the orthogonality between the FE and subscale spaces is guaranteed (and thus $(\partial_t \mathbf{u}_h, \tilde{\mathbf{u}}) = 0$) and standard mass matrices are used. This is an advantage from which we can take profit to build more efficient solvers. In the numerical tests we will favour this alternative.

Neglecting the pressure subscales the OSS method is given by (21) with $B = B_{oss}$ and $L_h(\mathbf{v}_h, q_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle$ where

$$
\begin{aligned}
B_{oss}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = {} & B(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) \\
& + (\tau_m(-\nu\Delta\mathbf{u}_h + \mathbf{u}_h \cdot \nabla\mathbf{u}_h + \nabla p_h), \nu\Delta\mathbf{v}_h + \mathbf{u}_h \cdot \nabla\mathbf{v}_h + \nabla q_h) \\
& + (\tau_m \boldsymbol{\eta}_h, \nu\Delta\mathbf{v}_h + \mathbf{u}_h \cdot \nabla\mathbf{v}_h + \nabla q_h).
\end{aligned}
\tag{28}
$$

where $\boldsymbol{\eta}_h := \Pi_h(\mathbf{R}_u)$ is computed solving

$$
(\tau_m \boldsymbol{\eta}_h, \boldsymbol{\kappa}_h) + (\tau_m(-\nu\Delta\mathbf{u}_h + \mathbf{u}_h \cdot \nabla\mathbf{u}_h + \nabla p_h), \boldsymbol{\kappa}_h) = (\tau_m \mathbf{f}, \boldsymbol{\kappa}_h) \qquad \forall \boldsymbol{\kappa}_h \in \mathcal{V}_{h,0},
\tag{29}
$$

An implicit implementation of this method would require the introduction of an extra variable (the projection $\boldsymbol{\eta}_h$) and, more importantly, this variable would be coupled with both velocity and pressure. Because the sign of the Laplacian term multiplying the projection in (28) and the sign of the Laplacian term in (29) are different, the velocity-projection coupling is non-symmetric.

In order to reduce the coupling between variables we consider the *term by term* OSS method proposed in [19]. The main goal of this alternative is to stabilize separately the convective term and the pressure gradient term by two uncoupled orthogonal projections. As noted in [19] the *term by term* OSS has better stability properties than the classical one. Considering quasi-static and linear subscales the discrete problem is given by (21) with $B = B_{tbt\_oss}$ and $L_h(\mathbf{v}_h, q_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle$ where

$$
\begin{aligned}
B_{tbt\_oss}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = {} & B(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) \\
& + (\tau_m \mathbf{u}_h \cdot \nabla\mathbf{u}_h, \mathbf{u}_h \cdot \nabla\mathbf{v}_h) + (\tau_m \nabla p_h, \nabla q_h) \\
& - (\tau_m \boldsymbol{\eta}_h, \mathbf{u}_h \cdot \nabla\mathbf{v}_h) - (\tau_m \boldsymbol{\xi}_h, \nabla q_h).
\end{aligned}
\tag{30}
$$

and $\boldsymbol{\eta}_h := \Pi_h(\mathbf{u}_h \cdot \nabla\mathbf{u}_h)$ and $\boldsymbol{\xi}_h := \Pi_h(\nabla p_h)$ are computed solving

$$
\begin{aligned}
(\tau_m \boldsymbol{\eta}_h, \boldsymbol{\kappa}_h) &= (\tau_m \mathbf{u}_h \cdot \nabla\mathbf{u}_h, \boldsymbol{\kappa}_h) & \forall \boldsymbol{\kappa}_h \in \mathcal{V}_{h,0}, \\
(\tau_m \boldsymbol{\xi}_h, \boldsymbol{\zeta}_h) &= (\tau_m \nabla p_h, \boldsymbol{\zeta}_h) & \forall \boldsymbol{\zeta}_h \in \mathcal{V}_{h,0}.
\end{aligned}
\tag{31}
\tag{32}
$$

Note that with the *term by term* OSS method with implicit FE projections, there are $3d+1$ unknowns per node, while for ASGS the number of unknowns is $d+1$ per node. A priory it seems that such increase of unknowns make the former method not appealing in front of ASGS, but we will see later that the increase of computational cost is not linear with the increase of unknowns in this case. Furthermore, an optimal block preconditioning technique can be used to solve problem (21)-(30)-(31)-(32) taking advantage of its block structure. This point is further discussed in Sect. 4.

3.3. Term by term OSS with ISS elements

For equal order interpolation the pressure stabilization is mandatory but for ISS FE spaces pressure stability is guaranteed. Then, we could define a *term by term* OSS method for ISS FE that only stabilizes the convective term by an orthogonal FE projection. This approach would reduce the number of unknowns per node, keeping stability properties and the better conditioned matrix than ASGS. The definition of the *term by term* OSS-ISS method can be given by equation (21) with $B = B_{tbt\_oss\_iss}$ and $L_h(\mathbf{v}_h, q_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle$ where

$$
\begin{aligned}
B_{tbt\_oss\_iss}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = {} & B(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) \\
& + (\tau_m \mathbf{u}_h \cdot \nabla\mathbf{u}_h, \mathbf{u}_h \cdot \nabla\mathbf{v}_h) - (\tau_m \boldsymbol{\eta}_h, \mathbf{u}_h \cdot \nabla\mathbf{v}_h) \\
& + (\tau_c \nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h),
\end{aligned}
\tag{33}
$$

complemented by the FE projection of the convective term defined in (31). Note that in this case we include the effect of the pressure subscales, that is, the grad-div stabilization. However (for constant stabilization parameters) the projection of the divergence required to implement (15) needs not to be computed because it vanishes, which is implied by the discrete mass conservation equation. This is not the case when pressure stabilization is used. When τ_c is variable we neglect this projection to reduce the computational cost. Note, however, that this approximation does not introduce any consistency error.

3.4. The grad-div stabilization

It is important to highlight here the presence of the pressure subscale term, $\tau_c(\nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)$ in (33), first introduced in [28]. Even though it is usually included in VMS formulations it is sometimes neglected in practice, particularly when equal order interpolations with pressure stabilization are considered [16]. As it is shown in [23], using the grad-div stabilization in the simulation of laminar flows produces a *small improvement* of the results obtained with equal order interpolation but a *clear improvement* of the results obtained with ISS elements.

In the VMS decomposition, this term comes from the residual of the incompressibility constraint and its addition improves the conservation of mass as well as the effect that the error on the pressure field produces on the velocity field. In [7, 29] the authors assessed the use of ISS elements for the incompressible Navier-Stokes equations in the laminar regime, highlighting the importance of this term. The optimal choice of the parameter τ_c, discussed in [30], depends on the relative norms of the velocity and pressure (is therefore problem dependent) and can be of order one but also much bigger. On the other hand, in [31] it is proved that on a mesh that is a barycentre-refinement of a regular mesh , the Taylor-Hood (TH) approximations converge to a point-wise divergence-free solution, the one obtained using Scott-Vogelius (SV) elements [32], as $\tau_c \to \infty$. Then, it is seen that the optimal value of τ_c is an open question and, as stated in [7], *we may consider the search of optimal parameters as a trade-off between mass and momentum balance in the FE system.* In this work we will try to evaluate the importance of such term for turbulent incompressible flows when ISS elements are used. Thus, a detailed discussion of which are the values that should take τ_c is considered for each numerical test in further sections.

Apart from its influence on mass conservation, this term is also known to introduce numerical dissipation both when equal order [16] or ISS elements are used [7]. The energy balance of the ASGS method is quite complex (because it is not purely dissipative). It is discussed in [33, 16]. The energy balance of the OSS method is obtained by taking $v_h = u_h$ and $q_h = p_h$ in (21), (31) and (32), and using (30), getting:

$$\frac{1}{2}\frac{d}{dt}\|\mathbf{u}_h\|^2 + \nu\|\nabla\mathbf{u}_h\|^2 + \|\tau_m^{1/2}(\mathbf{u}_h \cdot \nabla\mathbf{u}_h - \boldsymbol{\eta}_h)\|^2 + \|\tau_m^{1/2}(\nabla p_h - \boldsymbol{\xi}_h)\|^2 + \|\tau_c^{1/2}\nabla \cdot \mathbf{u}_h\|^2 = \langle \mathbf{f}, \mathbf{u}_h \rangle. \quad (34)$$

Apart from the viscous dissipation coming from the Galerkin method, which is negligible in turbulent flows, we get extra dissipation that comes from the control of the orthogonal projection of the convective term. We also have the pressure dissipation and the dissipation that comes from the grad-div stabilization. When equal order interpolations are used, the fourth (dissipative) term permits to control the (otherwise unstable) pressure and the extra dissipation of the grad-div term is not necessary; very good results are obtained taking $\tau_c = 0$ as shown in [16]. On the other hand, for ISS discretizations, the numerical dissipation introduced by the grad-div term is crucial to obtain accurate solutions. For instance, given a velocity space, the TH element has one order less pressure space than the stabilized equal order counterpart, leading to a poorer approximation of the mass conservation equation. Note that when τ_c goes to zero no dissipation is introduced but the accuracy in the satisfaction of mass conservation is poor. On the other hand taking τ_c large results in a strong imposition of the mass conservation (it acts as a penalty term) giving, for the TH pair on a mesh that is a barycentre-refinement of a regular mesh, exact (point-wise) zero divergence in the limit [31]. In this case it turns out that the extra dissipation also vanishes as $\|\tau_c^{1/2}\nabla \cdot \mathbf{u}_h\| \longrightarrow 0$ in the limit of $\tau_c \longrightarrow \infty$ as $\tau_c^{-1/2}$ [34]. We stress that this is not true for general meshes. Clearly, wrong results can be obtained in this limit [31]. In any case it is important to keep in mind that when τ_c is changed the velocity field changes and the other dissipative terms in (34) also change. We cannot therefore conclude whether the method is more or less dissipative looking only at $\|\tau_c^{1/2}\nabla \cdot \mathbf{u}_h\|^2$. It is possible to arrive to such a conclusion when this parameter does not influence very much the solution, as in the case of equal order interpolation. This point is discussed in detail when presenting the numerical results in Sect. 5.

Finally, the grad-div term also has a strong influence on the conditioning of the linear system and therefore on the convergence of iterative solvers. When the monolithic system is considered, this term acts as an augmented Lagrangian term improving the convergence of block iterative schemes but it is also known that it introduces stiffness in the velocity block [24, 25].

An alternative to the parameter *tuning* needed for, e.g., Taylor Hood elements, is the use of divergence-free FEs that also satisfy the inf-sup condition (obviously, for these elements the grad-div stabilization vanishes). One of this group of elements is the SV pair [32] which is given by the triangular/tetrahedral elements P_k/P_{k-1}^{disc}. This element is similar to the TH element except that the pressure space is discontinuous, which implies the property of point-wise divergence-free (taking the pressure test function as the divergence of the velocity). The SV element is ISS under certain assumptions on the mesh, e.g., the order of the interpolation $k \geq d$ and the mesh is a barycentre-refinement of a regular mesh [34]. For quadrilateral meshes, Zhang introduced a new divergence-free ISS element in [35]. This work proposed a new family of quadrilateral elements that have a different interpolation space for each velocity component, which for the 3D case can be stated as $Q_{k+1,k,k} \times Q_{k,k+1,k} \times Q_{k,k,k+1}/Q_k^{disc}$. In this case, the pressure field is also discontinuous with spurious modes filtered. We do not consider these approaches here.

4. Block preconditioning for the monolithic problem

A common approach when the OSS method is used is to treat the projection within a Picard linearization scheme. That means to compute its value after the resolution of the velocity-pressure system and iterating until the solution converges. Although the resulting matrix has a better condition number than the ASGS method, the increase of nonlinear iterations due to the explicit treatment of the orthogonal projection may cause a lose of efficiency of this method in many cases. In [16] there is a computational cost analysis of these methods for turbulent incompressible flows where this effect can be seen.

Alternatively, the implicit approach of the OSS method increases the number of unknowns of the problem, not only having the usual velocity and pressure unknowns, but also the FE projection. Treating the FE projection as a new unknown, the system of equations to be solved is increased with equation (24) or (25). This projection unknown is coupled with both the velocity and the pressure and these blocks are non-symmetric which makes the application of the block preconditioning technique more difficult. As mentioned, this is not the case when the term by term OSS or the term by term OSS with ISS elements are considered.

In the numerical experiments of Section 5.2 and Section 5.3 a Crank-Nicolson time integration scheme is used whereas in the numerical experiments of Section 5.1 the steady state problem is directly solved. In both cases the Picard method is used to solve the nonlinear problem using as convergence criterium that the increment (in Euclidean norm) of the unknown vector is less than 10^{-5}. At each nonlinear iteration we solve the monolithic linear system (defined in (35) below) using the GMRES method with a left block preconditioner described below; convergence is achieved when the ration between the Euclidean norm of the final and initial residual is less than 10^{-7}. We consider the implicit term-by-term OSS stabilization given by the bilinear form (30) as this is the more general case when talking about number of unknowns that appear in the system, since there are the velocity, pressure and two projections. The case of the term by term OSS with ISS elements and the ASGS are obtained just eliminating rows and columns of this system.

Assuming that $\mathbf{u}_h$, p_h, $\boldsymbol{\eta}_h$ and $\boldsymbol{\xi}_h$ are defined by a FE interpolation from the nodal values $\{\mathbf{U}^a\}_{a=1,...,N_u}$, $\{P^b\}_{b=1,...,N_p}$, $\{\boldsymbol{\Upsilon}^l\}_{l=1,...,N_\eta}$ and $\{\boldsymbol{\Xi}^m\}_{m=1,...,N_\xi}$, the FE approximation of the velocity, pressure and projection fields can be written as

$$\mathbf{u}_h(x) = \sum_{a=1}^{N_u} \boldsymbol{\phi}_a(x)\mathbf{U}^a, \quad p_h(x) = \sum_{b=1}^{N_p} \psi_b(x)P^b, \quad \boldsymbol{\eta}_h(x) = \sum_{l=1}^{N_\eta} \boldsymbol{\phi}_\eta(x)\boldsymbol{\Upsilon}^l, \quad \boldsymbol{\xi}_h(x) = \sum_{m=1}^{N_\xi} \boldsymbol{\phi}_\xi(x)\boldsymbol{\Xi}^m,$$

where $\{\boldsymbol{\phi}_{a,i}\}_{a=1,...,N_u;i=1,...,d}$, $\{\psi_b\}_{b=1,...,N_p}$, $\{\boldsymbol{\phi}_{l,i}\}_{l=1,...,N_\eta;i=1,...,d}$ and $\{\boldsymbol{\phi}_{m,i}\}_{m=1,...,N_\xi;i=1,...,d}$ are the Lagrangian basis associated to $\mathcal{V}_h$ and $\mathcal{Q}_h$. N_u, N_p, N_η and N_ξ are the total amount of nodes for the velocity, pressure and projection fields.

The matrix form of the problem (21) with the bilinear form (30) and, the projections (31)-(32) can

be written as

$$
\begin{bmatrix}
\frac{1}{\delta t}\mathbb{M} + \mathbb{K} + \mathbb{C} + \mathbb{A}_\tau & \mathbb{G} & \mathbb{B}_{\eta,\tau} & 0 \\
\mathbb{D} & \mathbb{L}_\tau & 0 & \mathbb{B}_{\xi,\tau} \\
-\mathbb{B}_{\eta,\tau}^T & 0 & \mathbb{M}_{\eta,\tau} & 0 \\
0 & -\mathbb{B}_{\xi,\tau}^T & 0 & \mathbb{M}_{\xi,\tau}
\end{bmatrix}
\begin{bmatrix}
\mathbf{U} \\ \mathbf{P} \\ \boldsymbol{\Upsilon} \\ \boldsymbol{\Xi}
\end{bmatrix}
=
\begin{bmatrix}
\mathbf{F}_u \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0}
\end{bmatrix},
\tag{35}
$$

where $\mathbb{M}$, $\mathbb{K}$, $\mathbb{C}$, $\mathbb{D}$ and $\mathbb{G}$ are the matrices that arise from the Galerkin integration of the mass, diffusive, convective, velocity divergence and pressure gradient terms, respectively. The definition of the remaining terms are given by

$$
\begin{aligned}
\mathbb{A}_\tau^{ab} &:= (\tau_m \mathbf{u} \cdot \nabla \boldsymbol{\phi}_a, \mathbf{u} \cdot \nabla \boldsymbol{\phi}_b) + (\tau_c \nabla \cdot \boldsymbol{\phi}_a, \nabla \cdot \boldsymbol{\phi}_b), & a, b &= 1, ..., N_u, \\
\mathbb{L}_\tau^{ab} &:= (\tau_m \nabla \psi_a, \nabla \psi_b), & a, b &= 1, ..., N_p, \\
\mathbb{B}_{\eta,\tau}^{ab} &:= -(\tau_m \mathbf{u} \cdot \nabla \boldsymbol{\phi}_a, \boldsymbol{\phi}_b), & a &= 1, ..., N_u, b = 1, ..., N_\eta, \\
\mathbb{M}_{\eta,\tau}^{ab} &:= -(\tau_m \boldsymbol{\phi}_a, \boldsymbol{\phi}_b), & a, b &= 1, ..., N_\eta, \\
\mathbb{B}_{\xi,\tau}^{ab} &:= -(\tau_m \nabla \psi_a, \boldsymbol{\phi}_b), & a &= 1, ..., N_p, b = 1, ..., N_\xi, \\
\mathbb{M}_{\xi,\tau}^{ab} &:= -(\tau_m \boldsymbol{\phi}_a, \boldsymbol{\phi}_b), & a, b &= 1, ..., N_\xi,
\end{aligned}
$$

being a and b the node identification. Note that $\mathbb{D} = -\mathbb{G}^T$, when Dirichlet boundary conditions are considered. In general, $N_\xi = N_\eta$, then, $\mathbb{M}_{\xi,\tau} = \mathbb{M}_{\eta,\tau}$.

For the ASGS method, only the first two rows and columns of the matricial system (35) are present, with different definitions of $\mathbb{A}_\tau$ and $\mathbb{L}_\tau$ that are straight forward from the bilinear form (22). In the case of OSS-ISS method, only the last row and column disappear, keeping the same definition for the remaining terms.

To solve the system (35) we use a recursive block-preconditioning technique. This methodology was used in [26] for a multiphysics problem like the thermally coupled inductionless MHD. The idea is to construct recursively block preconditioners of size 2×2 from an incomplete block factorization of the original 2×2 block matrices.

Let us consider a block system equivalent to (35) defined as

$$
\begin{bmatrix}
\tilde{\mathbb{M}}_\tau & -\tilde{\mathbb{B}}_\tau^T \\
\tilde{\mathbb{B}}_\tau & \tilde{\mathbb{K}}_\tau
\end{bmatrix}
\begin{bmatrix}
\tilde{\boldsymbol{\Xi}} \\ \tilde{\mathbf{U}}
\end{bmatrix}
=
\begin{bmatrix}
\mathbf{0} \\ \tilde{\mathbf{F}}_u
\end{bmatrix},
\tag{36}
$$

with $\tilde{\mathbb{K}}_\tau := \begin{bmatrix} \frac{1}{\delta t}\mathbb{M} + \mathbb{K} + \mathbb{C} + \mathbb{A}_\tau & \mathbb{G} \\ \mathbb{D} & \mathbb{L}_\tau \end{bmatrix}$, $\tilde{\mathbb{B}}_\tau := \begin{bmatrix} \mathbb{B}_{\eta,\tau} & 0 \\ 0 & \mathbb{B}_{\xi,\tau} \end{bmatrix}$, $\tilde{\mathbb{M}}_\tau := \begin{bmatrix} \mathbb{M}_{\eta,\tau} & 0 \\ 0 & \mathbb{M}_{\xi,\tau} \end{bmatrix}$, $\tilde{\boldsymbol{\Xi}} := \begin{bmatrix} \boldsymbol{\Upsilon} \\ \boldsymbol{\Xi} \end{bmatrix}$,
$\tilde{\mathbf{U}} := \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix}$ and $\tilde{\mathbf{F}}_u := \begin{bmatrix} \mathbf{F}_u \\ 0 \end{bmatrix}$.

The matrix that appear in the system of equations (36) can be factorized into an exact block *LU* matrix product as follows

$$
\begin{aligned}
\tilde{\mathbb{A}} &:= \begin{bmatrix} \tilde{\mathbb{M}}_\tau & -\tilde{\mathbb{B}}_\tau^T \\ \tilde{\mathbb{B}}_\tau & \tilde{\mathbb{K}}_\tau \end{bmatrix} = \begin{bmatrix} \tilde{\mathbb{M}}_\tau & 0 \\ \tilde{\mathbb{B}}_\tau & \tilde{\mathbb{S}} \end{bmatrix} \begin{bmatrix} \mathbb{I} & -\tilde{\mathbb{M}}_\tau^{-1}\tilde{\mathbb{B}}_\tau^T \\ 0 & \mathbb{I} \end{bmatrix} \\
&= \begin{bmatrix} \mathbb{I} & 0 \\ \tilde{\mathbb{B}}_\tau \tilde{\mathbb{M}}_\tau^{-1} & \mathbb{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbb{M}}_\tau & -\tilde{\mathbb{B}}_\tau^T \\ 0 & \tilde{\mathbb{S}} \end{bmatrix},
\end{aligned}
\tag{37}
$$

being $\tilde{\mathbb{S}} := \tilde{\mathbb{K}}_\tau + \tilde{\mathbb{B}}_\tau^T \tilde{\mathbb{M}}_\tau^{-1} \tilde{\mathbb{B}}_\tau$ the Schur complement with respect to $\mathbf{U}$. In order to construct a preconditioner to the system (36) we build an inexact factorization of $\tilde{\mathbb{A}}$ such that each diagonal block is recursively preconditioned by another block preconditioner. We consider three different preconditioners of $\tilde{\mathbb{A}}$, which try to approximate the *LU* decompositions (37)

Diagonal preconditioner (D):
$$
P_D(\tilde{\mathbb{A}}) = \begin{bmatrix} \tilde{\mathbb{M}}_\tau & 0 \\ 0 & \tilde{\mathbb{S}} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbb{M}}_\tau^{-1} & 0 \\ 0 & \tilde{\mathbb{S}}^{-1} \end{bmatrix},
\tag{38}
$$

Upper preconditioner (U):
$$
P_U(\tilde{\mathbb{A}}) = \begin{bmatrix} \tilde{\mathbb{M}}_\tau & -\tilde{\mathbb{B}}_\tau^T \\ 0 & \tilde{\mathbb{S}} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbb{M}}_\tau^{-1} & \tilde{\mathbb{M}}_\tau^{-1}\tilde{\mathbb{B}}_\tau^T\tilde{\mathbb{S}}^{-1} \\ 0 & \tilde{\mathbb{S}}^{-1} \end{bmatrix},
\tag{39}
$$

Lower preconditioner (L):
$$
P_L(\tilde{\mathbb{A}}) = \begin{bmatrix} \tilde{\mathbb{M}}_\tau & 0 \\ \tilde{\mathbb{B}}_\tau & \tilde{\mathbb{S}} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbb{M}}_\tau^{-1} & 0 \\ -\tilde{\mathbb{S}}^{-1}\tilde{\mathbb{B}}_\tau\tilde{\mathbb{M}}_\tau^{-1} & \tilde{\mathbb{S}}^{-1} \end{bmatrix}.
\tag{40}
$$

To apply (38)-(40) we need to compute the action of $\tilde{\mathbb{M}}_\tau^{-1}$ and $\tilde{\mathbb{S}}^{-1}$. In the first case, $\tilde{\mathbb{M}}_\tau^{-1}$ is block diagonal thus requiring the inverse of each projection mass matrix, $\mathbb{M}_{\eta,\tau}^{-1}$ and $\mathbb{M}_{\xi,\tau}^{-1}$. In a serial computation, they are applied by a direct method whereas in a parallel computation they are approximated by one application of a diagonal Domain Decomposition (DD) preconditioner constructed with the values of the diagonal of $\mathbb{M}_{\eta,\tau}$ and $\mathbb{M}_{\xi,\tau}$, respectively. In the second case we approximate the Schur complement inverse as $\tilde{\mathbb{S}} \approx \tilde{\mathbb{K}}_\tau$, that is, we neglect the contribution from the projections $\tilde{\mathbb{B}}_\tau^T \tilde{\mathbb{M}}_\tau^{-1} \tilde{\mathbb{B}}_\tau$ in the preconditioner. This approximation corresponds to using as a preconditioner the solution of a system arising from a non-consistent formulation in which artificial pressure and streamline diffusion are added. Positivity of the preconditioner is therefore guaranteed.

For the application of $\tilde{\mathbb{K}}_\tau^{-1}$ we consider other three preconditioners equivalent to (38)-(40). Using the notation $\mathbb{K}_\tau := \frac{1}{\delta t}\mathbb{M} + \mathbb{K} + \mathbb{C} + \mathbb{A}_\tau$ for the velocity block matrix and $\mathbb{S} : \mathbb{L}_\tau - \mathbb{D}\mathbb{K}_\tau^{-1}\mathbb{G}$ for the Schur complement associated to the pressure field, we have the following block preconditioners

Diagonal preconditioner (D): $\qquad P_D(\tilde{\mathbb{K}}_\tau) = \begin{bmatrix} \mathbb{K}_\tau & 0 \\ 0 & \mathbb{S} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbb{K}_\tau^{-1} & 0 \\ 0 & \mathbb{S}^{-1} \end{bmatrix},$ (41)

Upper preconditioner (U): $\qquad P_U(\tilde{\mathbb{K}}_\tau) = \begin{bmatrix} \mathbb{K}_\tau & \mathbb{G} \\ 0 & \mathbb{S} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbb{K}_\tau^{-1} & -\mathbb{K}_\tau^{-1}\mathbb{G}\mathbb{S}^{-1} \\ 0 & \mathbb{S}^{-1} \end{bmatrix},$ (42)

Lower preconditioner (L): $\qquad P_L(\tilde{\mathbb{K}}_\tau) = \begin{bmatrix} \mathbb{K}_\tau & 0 \\ \mathbb{D} & \mathbb{S} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbb{K}_\tau^{-1} & 0 \\ -\mathbb{S}^{-1}\mathbb{D}\mathbb{K}_\tau^{-1} & \mathbb{S}^{-1} \end{bmatrix}.$ (43)

In this case, the inverse of the Schur complement is approximated by a pressure Laplacian matrix, $\mathbb{S}^{-1} \sim \delta t \mathbb{L}_p^{-1}$. In regimes where the viscous term becomes dominant, one can use a Cahouet-Chabard type preconditioner to approximate the inverse of the Schur complement, i.e., $\mathbb{S}^{-1} \sim (\nu+\tau_c)\mathbb{M}_p^{-1}+\delta t\mathbb{L}_p^{-1}$, see [25, 22]. For the serial case, the Laplacian matrix is inverted using a direct method, but in a parallel context we approximate the inverse of $\mathbb{L}_p$ by one application of a Balancing Domain Decomposition by Constraints (BDDC) preconditioner over such matrix. Something similar is done for the inverse of the velocity block matrix $\mathbb{K}_\tau$. Here we use a direct method for the serial case and one application of BDDC preconditioner when parallel solvers are treated. We refer to [36] for the application of the BDDC preconditioner to nonsymmetric problems. For a more detailed description of the implementation and algorithms used for the recursive block-preconditioning technique we refer to [26]. Besides monolithic approaches, splitting velocity-pressure techniques are commonly used for high Reynolds turbulent flows, which can be interpreted as one application of a block-preconditioner [37, 38]. High order time integration schemes that segregate velocity and pressure computation have recently been proposed in [39]. Since the stabilized inf-sup stable FEs proposed herein do not introduce any term at the pressure-pressure block, they are particularly well-suited for this type of schemes, because it keeps the index-2 differential-algebraic nature of the problem.

5. Numerical tests

In this section we provide some numerical results for turbulent incompressible flow simulations using all the methods stated above. First, we do a comparative analysis between the three stabilization methods developed in Sect. 3. The parameter election for the pressure subscale term is assessed in all numerical tests performed in this section. But, first of all, we show some results about the computational cost of the VMS methods considered in this work. To check the efficiency of each method we solve a simple 2D steady problem with analytical solution.

5.1. Analytical colliding flow

In this section we solve a problem with analytical solution that models a colliding flow. This test have been used in [37] to discuss error estimates for the Stokes and Navier-Stokes problems using ISS FE velocity-pressure pairs. It has an analytical solution with the expression:

$$\mathbf{u}(x,y,t) = \begin{bmatrix} 20xy^3 \\ 5x^4 - 5y^4 \end{bmatrix},$$ (44)

$$p(x,y) = 60x^2y - 20y^3 + 40.$$ (45)

The problem is solved in the square domain $[-1, 1] \times [-1, 1]$ with a Reynolds number $Re = 25$. Here we analyze the solver iterations convergence in h for each preconditioner defined in Sect. 4 as well as the elapsed CPU time and the error on the velocity and pressure fields. Due to the simplicity of the test, we will solve the problem in serial using $Q1/Q1$ elements meshes for the ASGS and term-by-term OSS, and $Q2/Q1$ elements mesh for the OSS-ISS method.

We only show the results when using the lower-triangular preconditioner for the global matrix $P_L(\tilde{\mathbb{A}})$, defined in (40). The diagonal version $P_D(\tilde{\mathbb{A}})$ defined in (38) gives poorer performance and same results are obtained with the upper-triangular version $P_U(\tilde{\mathbb{A}})$ (39). For the velocity-pressure block, all preconditioners defined in (41)-(43) are considered.

In Fig. 1 we depict the solver iterations needed to solve the problem as well the total elapsed CPU time for different meshes composed by $4, 8, 16, 32$ or 64 elements per direction. When the OSS-ISS method is considered, as we use $Q2/Q1$ elements, the number of elements per direction is divided by two. The amount of solver iterations shown in Fig. 1(a) is computed adding all solver iterations needed for all nonlinear iterations. We see in this figure that the OSS-ISS method has an increase of solver iterations for the coarser mesh, compared with the other two methods. This behaviour is produced by the fact that for this case the mesh is too coarse, 2×2 $Q2/Q1$ elements, and the nonlinear iterations suffer a drastic increase. For finer meshes, the OSS-ISS method needs less solver iterations than the other two. We see that for the diagonal preconditioner $P_D(\mathbb{K}_\tau)$ the OSS-ISS is the only that scales when we refine the mesh. The upper and lower preconditioners, $P_U(\mathbb{K}_\tau)$ and $P_L(\mathbb{K}_\tau)$, have similar results with a slightly better performance of the upper version for all methods. The ASGS and OSS methods also need a similar number of iterations to solve the problem, being the OSS method a little bit over the ASGS.

Looking at the elapsed computational time, Fig. 1(b), we see that the OSS method is clearly more expensive, in terms of consumed time, than the other two methods. The ASGS method is slightly faster than the OSS-ISS, but the differences are almost negligible for the $P_U(\mathbb{K}_\tau)$ preconditioner. The fact that the most expensive method is the OSS method is justified by the number of Degrees Of Freedom (DOFs) that appear in the system of equations. As stated in Sect. 4, the OSS method has two vectorial unknowns more than ASGS method. On the other hand, the OSS-ISS method also has an additional vectorial unknown than ASGS, but the pressure field is approximated with half nodes. Without taking into account the boundary conditions, for the 4×4 elements mesh (2×2 for the OSS-ISS case), we would have 175 DOFs for the OSS method, 75 DOFs for the ASGS method and 109 DOFs for the OSS-ISS method. Then, we see that, although it has more DOFs, the OSS-ISS method is comparable in terms of computational time with the ASGS. Further, the number of DOFs is a wrong measure of the CPU cost. The most expensive problem in such simulations is the pressure Poisson equation, which in this case is smaller (one order less) for ISS elements. Further, the projection DOFs are almost for free, since they just involve a mass matrix solve. Further, the ISS method exhibit a clearly lower number of iterations.

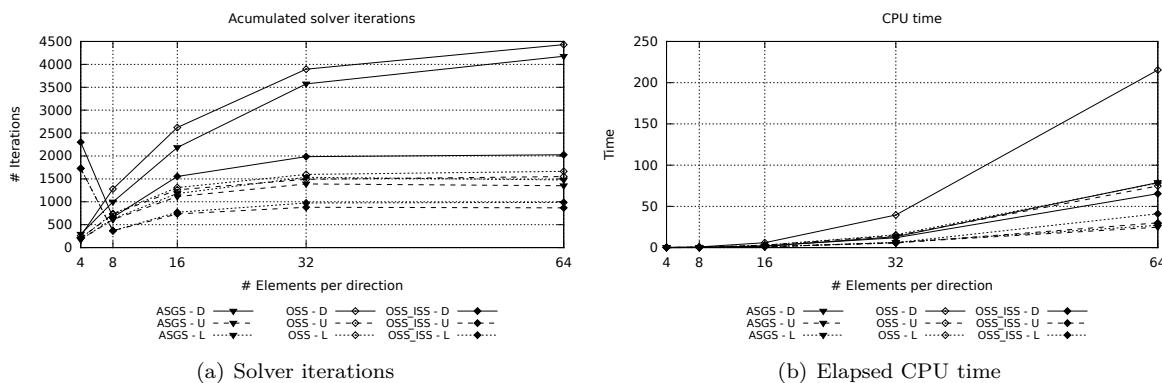(a) Solver iterations   (b) Elapsed CPU time

Figure 1: Colliding flow solver iterations and elapsed CPU time using $P_U(\tilde{\mathbb{A}})$ for the global matrix.

Let us now focus on the error in the velocity and pressure fields. Since we are solving a problem with analytical solution, we can evaluate exactly the error of the FE approximation, $e_u := \|\mathbf{u}_h - \mathbf{u}\|$

and $e_p := \|p_h - p\|$. Fig. 2 depicts the convergence of both errors when refining the mesh. In Fig. 2(a) we see that the velocity error converges as expected, with a 2nd order rate for the ASGS and OSS methods, which are approximated by $Q1/Q1$ elements, and with a 3rd order rate for the OSS-ISS method, which is approximated by a $Q2/Q1$ element. Looking at Fig. 2(b), it is seen that the pressure
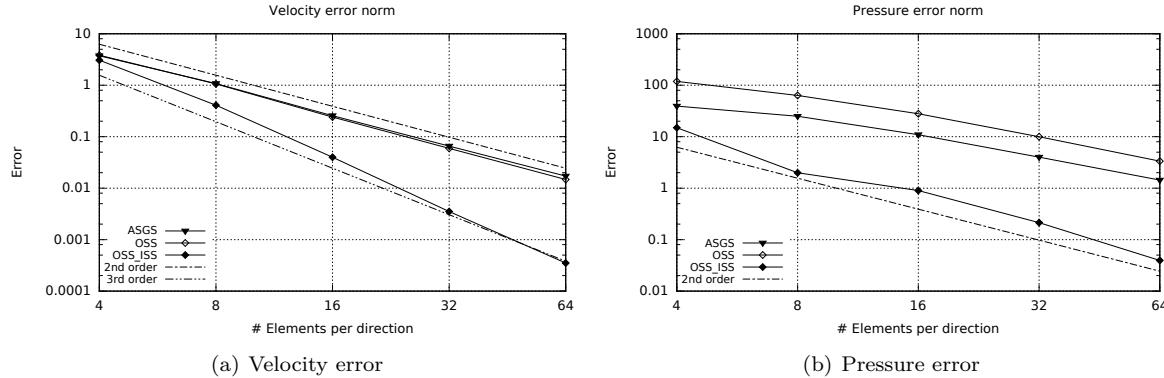

(a) Velocity error

(b) Pressure error

Figure 2: Colliding flow error convergence using $P_U(\tilde{\mathbb{A}})$ for the global matrix.

error norm converges with a 2nd order rate for all the methods, as expected. Note that here we only show the results obtained by using the diagonal preconditioner, since the solution is the same for all cases because we are not modifying the system that is being solved.

In order to check the efficiency of each method we compare the error norm with the elapsed computational time. In this way we have an idea of the time needed for a given method to achieve certain solution accuracy. Fig. 3 shows this comparison for velocity (Fig. 3(a)) and pressure (Fig. 3(b)) fields, where we see that, excluding the coarser mesh, the OSS-ISS method is much more efficient than the other two. Furthermore, the most efficient preconditioner is the upper version $P_U(\mathbb{K}_\tau)$.
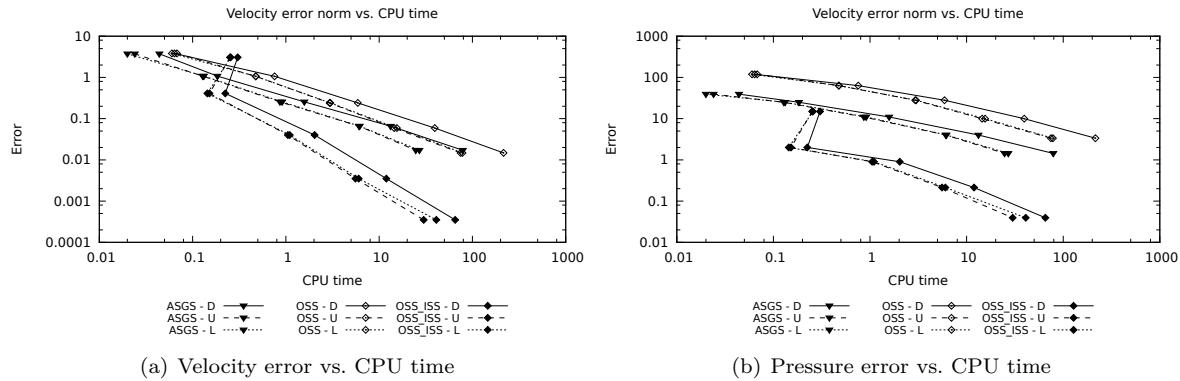

(a) Velocity error vs. CPU time

(b) Pressure error vs. CPU time

Figure 3: Colliding flow error vs. CPU time using $P_U(\tilde{\mathbb{A}})$ for the global matrix.

5.2. Taylor Green Vortex flow

It has been shown by many authors that VMS stabilization terms can act as a LES model for turbulent flows, introducing the appropriate dissipation of the small scales that are not captured by the coarse solution. In particular, ASGS and OSS methods were assessed in [16], showing that these methods are capable to perform a good LES simulation of different turbulent benchmark problems. In order to check the performance of the convection-only OSS stabilization of ISS elements for a LES simulation, we analyze its behaviour in the Taylor Green Vortex (TGV) flow. This problem is widely used for benchmarking turbulent numerical simulations.

5.2.1. Test setting

The computational domain is the cube $(0, 2\pi)^3$ with periodical boundary conditions and the initial condition is given by an analytical field (see, e.g., [40])

$$\mathbf{u}(x, y, z, 0) = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \begin{pmatrix} u_0 \cos(x) \sin(y) \sin(z) \\ -u_0 \sin(x) \cos(y) \sin(z) \\ 0 \end{pmatrix} \tag{46}$$

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

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 using a Reynolds number Re = 1600.

5.2.2. Comparison between different VMS methods

The problem is solved from $t = 0.0$ to $T = 10.0$ with a fixed time step size of $\delta t = 5.0 \cdot 10^{-2}$ using a Crank-Nicolson time integration scheme, and the results are compared against a DNS by Brachet et al. [40]. We discretize the domain using different choices of the number of elements and the order of approximation, having two different groups of discretizations; one with 32^3 velocity DOFs and another with 64^3 velocity DOFs (for each velocity component). The former will be composed by the following meshes: 32^3 $Q1/Q1$, 16^3 $Q2/Q2$ elements or 16^3 $Q2/Q1$ elements when we use ISS discretization. The second group of meshes is made by: 64^3 $Q1/Q1$, 32^3 $Q2/Q2$ or 32^3 $Q2/Q1$ elements. For the stabilized formulations, ASGS and OSS, the algorithmic constants are $c_1 = 12$ $c_2 = 2$ and $c_c = 0.0$, and for the OSS-ISS method the same c_1 and c_2 algorithmic constants are used, but $c_c = 4.0$ unless noted otherwise. This choice of c_c for the OSS-ISS method is assessed in a following subsection.

(a) Energy evolution
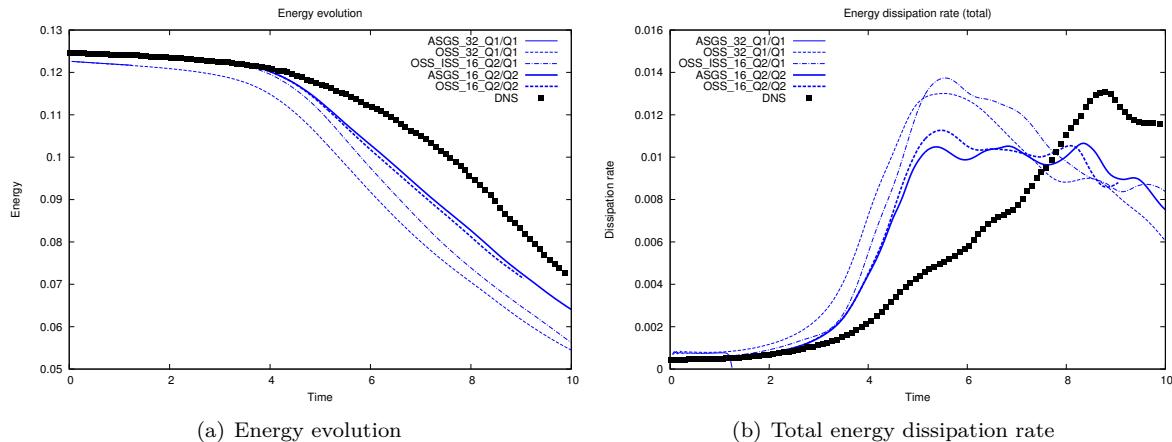
(b) Total energy dissipation rate

Figure 4: Energy and Total energy dissipation rate evolution with 32^3 velocity DOFs

In Fig. 4 we show the energy evolution and the energy dissipation rate for the ASGS, the OSS, and the OSS-ISS methods. A first thing that we have to state at this point is that the ASGS method with 32^3 $Q1/Q1$ has failed to converge at early stages of the problem, a behaviour also observed in [16]. Looking at Fig. 4(a) it is clear that the degree of interpolation makes a great difference on the solution, even with the same number of DOFs, the solution is more accurate when a higher order of interpolation is used. In the same figure we see that the loss of precision in the pressure for the ISS elements $Q2/Q1$ does affect the solution, giving a result between the $Q1/Q1$ and $Q2/Q2$ solutions. At Fig. 4(b) we see that the OSS-ISS method is more dissipative than the others. This behaviour is caused by the lack of accuracy in the pressure field, which in turn affects the conservation of mass of the problem.

13

Although an important improvement of the solution is achieved by increasing the order of interpolation, 32^3 velocity DOFs is still a very coarse mesh and the results shown in Fig. 4 are far from the DNS ones. Then, the same problem is solved in a finer mesh, with the double of velocity DOFs per direction, with the results shown in Fig. 5. In this case, all methods converge at all time steps. Fig. 5(a) depicts the energy evolution and it is also seen that the increase on the degree of interpolation results in a more accurate solution. Both ASGS and OSS methods with 32^3 $Q2/Q2$ elements have very accurate results, providing a solution almost on top of the DNS. There are very little differences between stabilization methods when the same discretization is used. Furthermore, the results of the OSS-ISS method are closer to the $Q1/Q1$ discretization than to the $Q2/Q2$ one. The total energy dissipation rate shown in Fig. 5(b) denote a very good agreement of the $Q2/Q2$ solution with the DNS, while the $Q1/Q1$ discretization for both ASGS and OSS methods are still more diffusive. Note that the OSS-ISS method for this discretization has more or less the same energy dissipation as the $Q1/Q1$ discretizations.

(a) Energy evolution
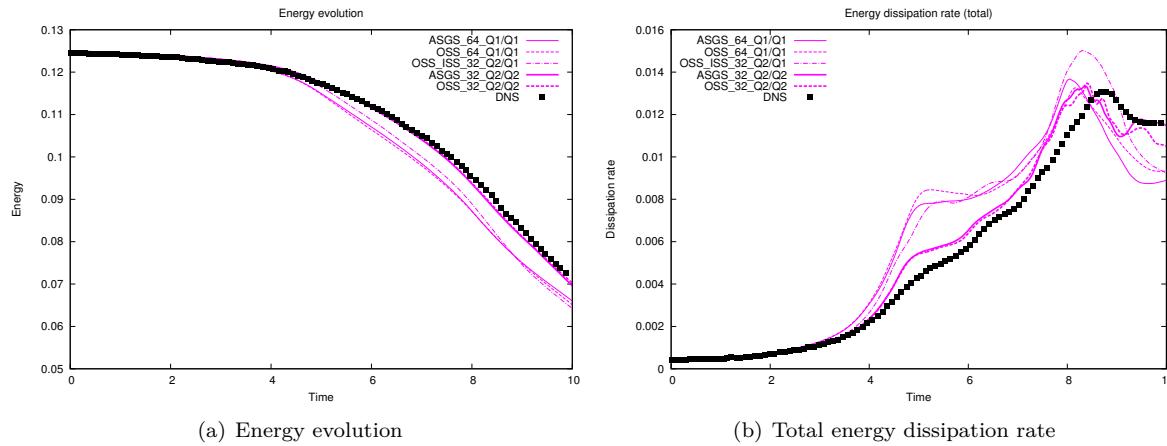
(b) Total energy dissipation rate

Figure 5: Energy and Total energy dissipation rate evolution with 64^3 velocity DOFs

When analyzing the suitability of a LES model, a very important turbulent quantity to take into account is the energy spectra. It gives us information about how the energy is distributed among the scales of the problem. In order to assess the behaviour of the proposed methods in that aspect, we compare our results by the DNS by Gassner et al. [41]. In Fig. 6 the energy spectra at $t = 9.0$ is depicted for both discretization groups. The coarser cases shown in Fig. 6(a) are all far from the DNS result, but follow the same pattern, with most of the energy on the greatest scales and little energy on the small scales, without any pileup of energy on the small scales. When the mesh is refined, see Fig. 6(b), the computed energy spectra tend to the DNS one. Note that also in this plot, $Q2/Q2$ discretization have better agreement with the DNS, specially on the small scales, where the influence of the enrichment of the interpolation space is patent.

5.2.3. Computational cost

It is clear that a good LES model has to reproduce as accurately as possible all the turbulent quantities, even for coarse meshes. The results presented until now show that ASGS and OSS methods perform better than the OSS-ISS when $Q2/Q2$ elements are used. Additionally, it has been seen that all methods converge to the DNS results when the mesh is refined. But a crucial point that has to be always taken into account when we talk about numerical simulations is the computational cost. At the end, we are looking for the cheapest method that allow us to reproduce accurately the physical phenomena that takes place in a turbulent flow. So in this subsection we will discuss the computational cost associated to each method and their efficiency when solving this kind of flows.

To check the computational cost we look at the number of solver iterations needed for each method, shown in Fig. 7. In particular, Fig. 7(a) depicts the total amount of solver iterations at each time step, adding up all nonlinear iterations. Note that the change of the nonlinear iterations along the time can be clearly noticed by the jumps on the curves. We see that the cheapest method is the OSS-ISS for

14

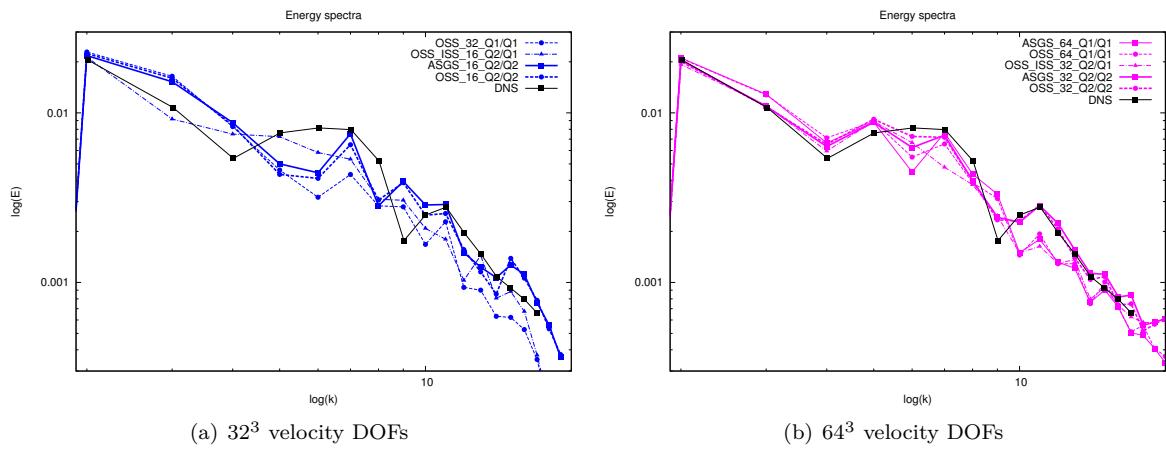(a) 32^3 velocity DOFs

(b) 64^3 velocity DOFs

Figure 6: Energy spectra at $t = 9.0$

both discretizations, 32^3 and 64^3 velocity DOFs, with much less solver iterations per time step than the other methods. It is also seen that ASGS is a little bit cheaper than the OSS method, which can be caused by the size of the system, much bigger for the OSS case due to the implicit treatment of the projections. An interesting result also seen in 7(a) is the improvement on the computational cost when we go from $Q1/Q1$ to $Q2/Q2$ discretization, keeping constant the number of DOFs. Looking at

(a) Solver iterations per time step

(b) Acumulated solver iterations
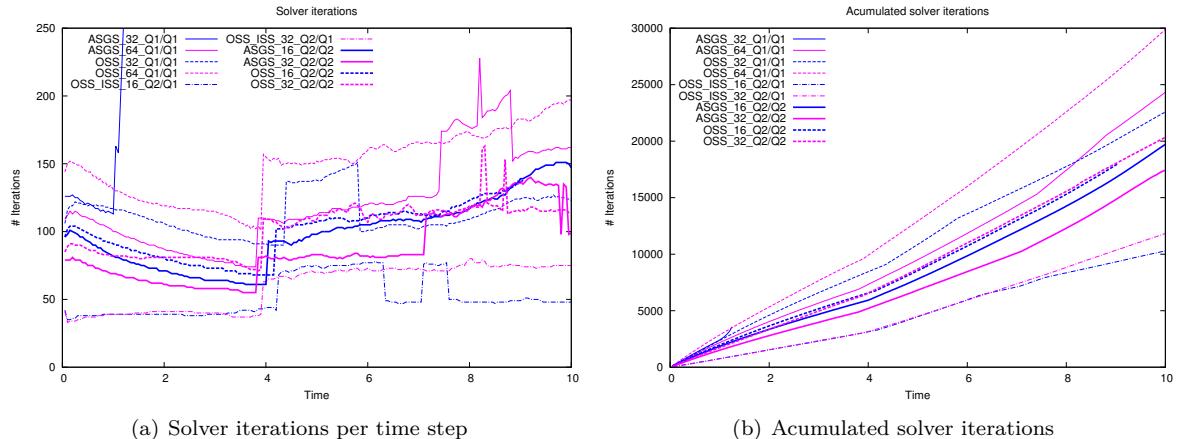
Figure 7: Computational cost

7(b) we realize that at the end of the computation, the total amount of solver iterations needed by OSS-ISS method is around 1.5 times less than the ASGS method, for the finer mesh, and half of them for the coarse mesh. This result indicates that the OSS-ISS is a very good approach as a LES model for turbulent problems since, although it has been seen that is not the most accurate when looking to the turbulent quantities, it is much cheaper and we can refine the mesh in order to get better results remaining competitive with the other methods. Further, let us remark that the pressure Poisson solvers are the most computationally intensive and this effect is more important for stabilized methods, in which there are more pressure degrees of freedom.

5.2.4. Influence of the pressure subscale term

As said above and also stated in Sect. 3.4, the pressure subscale $\tau_c(\nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)$ term has an important role when using ISS elements. In this subsection we are going to analyze the effect of this term on the results when simulating the TGV problem. Hence, we will consider the definition of τ_c given

15

in (20). We keep the algorithmic parameters $c1 = 12.0$ and $c2 = 2.0$ constant, so considering different values of c_c we can evaluate the influence of the pressure subscale term on the solution. In this case we choose six different configurations $c_c = \{0.0, 0.25, 0.5, 1.0, 2.0, 4.0\}$ and solve the problem with the 16^3 $Q2/Q1$ elements mesh. We compare the solution against the one obtained with the OSS method with 16^3 $Q2/Q2$ elements discretization.

In Fig. 8(a) the energy dissipation rate of the FE counterpart is shown. That is the viscous term $\nu\|\nabla\mathbf{u}_h\|^2$ that appears in equation (34). We see that when we reduce c_c the viscous dissipation introduced by the FE counterpart increase, being the case $c_c = 0.5$ the one closer to the DNS curve. But if we look at the energy spectra shown in Fig. 8(b), we see that this is not a good choice. What is actually happening when c_c goes to zero is that the dissipation is taking place in the largest scales of the problem, while the smallest ones keep the energy, resulting in an energy pileup at the tail of the spectra. This means that the energy is not dissipating in the correct way and the small scales, which are the ones that have more influence on the viscous dissipation term, have more energy than the desired one. Therefore, a good selection is to choose $c_c = 4.0$, which results are closer to the OSS method and has a better energy spectra shape. A higher value of c_c eventually leads to unstable solutions, for instance, with $c_c = 8.0$ and $\delta t = 5.0 \cdot 10^{-2}$ the solution fails to converge at $t = 0.4$. In that case, the solution becomes unstable and the nonlinear iteration does not reach the required tolerance.

(a) Viscous energy dissipation rate
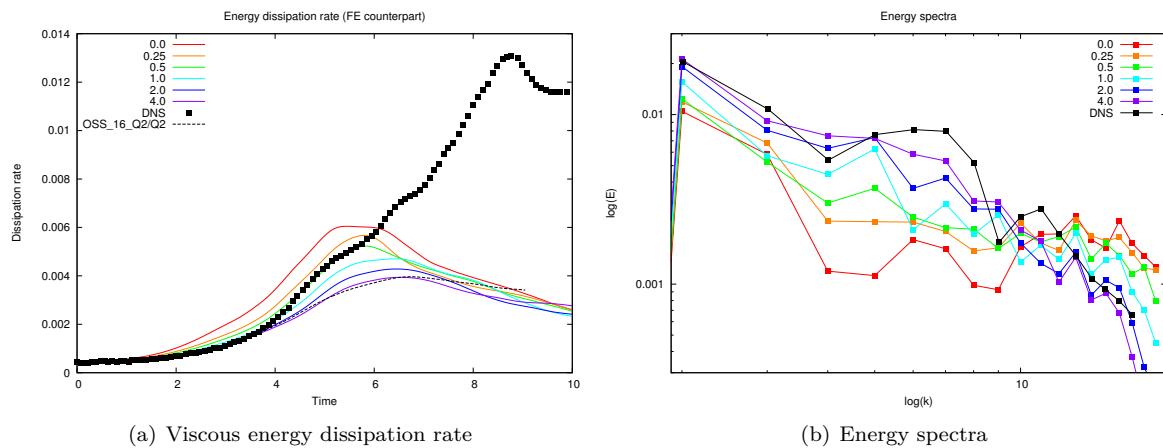
(b) Energy spectra

Figure 8: Comparison for different c_c choices for OSS-ISS with 32^3 velocity DOFs

As stated in Sect. 3.4, the pressure subscale term is essential to enforce the incompressibility constraint at the discrete level. In Fig. 9 the total energy dissipation rate (Fig. 9(a)) is depicted together with the velocity divergence $L2$-norm $\|\nabla \cdot \mathbf{u}_h\|$ (Fig. 9(b)). We see that, effectively, $\|\nabla \cdot \mathbf{u}_h\|$ is reduced when τ_c is increased. The case of $c_c = 0.0$ give especially bad results in terms of mass conservation, affecting also to the energy dissipation rate. In this case, as seen in Fig. 8(b), the over-dissipation on the large scales and the under-dissipation on the smallest ones make relevant the velocity spatial derivatives, thus increasing the $\|\nabla \cdot \mathbf{u}_h\|$ term. The introduction of $\tau_c\|\nabla \cdot \mathbf{u}_h\|$ into the energy dissipation equation (34) changes the way in which the flow dissipates its energy among the different scales, decreasing the energy of the small scales and, then, reducing the importance of the spatial derivatives, which is reflected in a lower value of $\|\nabla \cdot \mathbf{u}_h\|$ but also in a lower dissipation rate, as seen in Fig. 9(a), even if a positive term has been added to (34).

5.2.5. Refinement analysis for the OSS-ISS method

In the computational cost analysis shown above we have seen that OSS-ISS has a great potential as a LES model, especially when the computational cost is taken into account. This subsection aims to check the performance of this method when we refine the mesh both reducing the element size and increasing the interpolation order.

Once determined that the best algorithmic constant for the pressure subscale term is $c_c = 4.0$ when ISS FEs are used, we keep this value constant and change the discretization. A refinement analysis can be
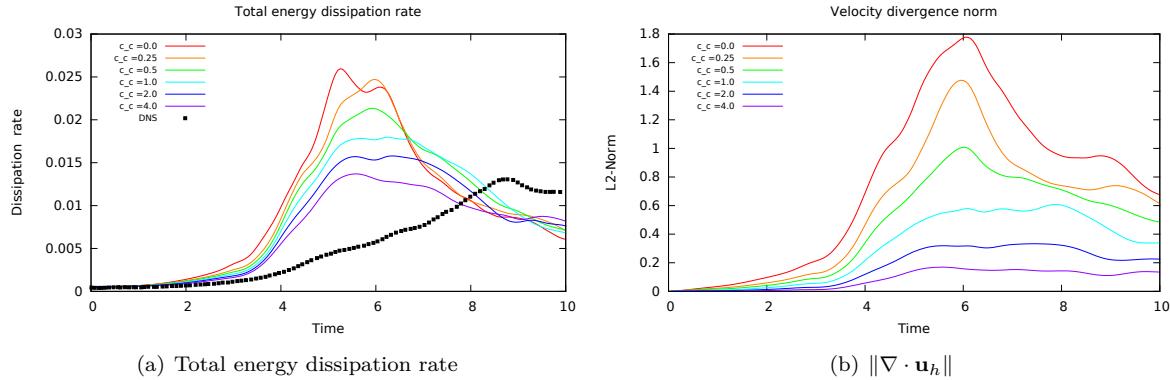
16

(a) Total energy dissipation rate

(b) $\|\nabla \cdot \mathbf{u}_h\|$

Figure 9: Total energy dissipation and velocity divergence $L2$-norm for different c_c choices for OSS-ISS with 32^3 velocity DOFs

done to determine if the LES approach presented in this work effectively converges to the DNS solution when we refine the mesh. In section 5.2 the TGV problem has been solved using different discretizations with 32^3 and 64^3 velocity DOFs. Now we go further and also solve the problem with 96^3 velocity DOFs, that is a 48^3 $Q2/Q1$ elements mesh. Furthermore, here we also use $Q3/Q2$ FEs. In particular, a 21^3 and 32^3 $Q3/Q2$ elements meshes are used, corresponding to the group of 64^3 and 96^3 velocity DOFs, respectively. We also decrease the time step to $\delta t = 2.5 \cdot 10^{-2}$ for the discretizations with 64^3 velocity DOFs and $\delta t = 1.5 \cdot 10^{-2}$ for the discretizations with 96^3 velocity DOFs.

In Fig. 10 we depict the kinetic energy and the total energy dissipation rate evolution for the different discretizations considered in this refinement analysis. Looking at the energy evolution in Fig. 10(a) it is clearly seen that the solution converge to the DNS results, giving the 32^3 $Q3/Q2$ elements mesh a very accurate solution, which is also evident in Fig. 10(b) where the total energy dissipation rate of this discretization is on top of the DNS solution. We also see in Fig. 10 the relevance of the degree of interpolation, where, for a given number of DOFs, the higher-order discretization results in a better solution.
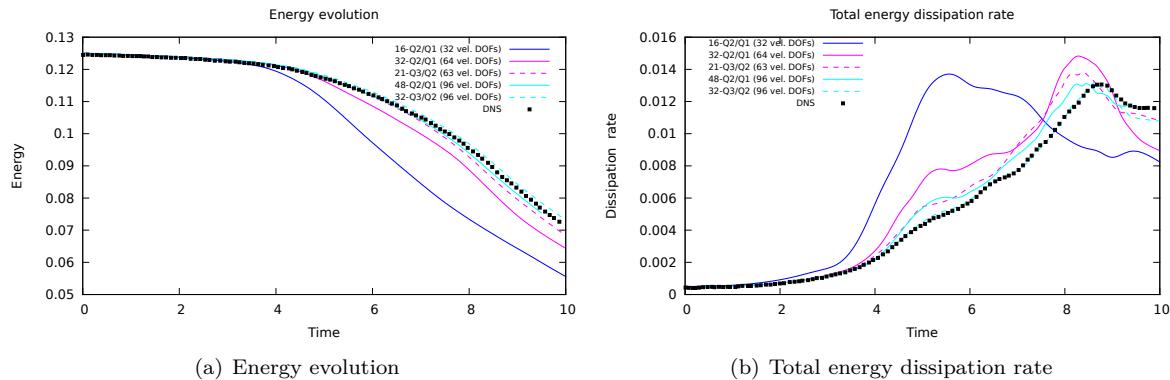

(a) Energy evolution

(b) Total energy dissipation rate

Figure 10: Energy and Total energy dissipation rate evolution refining the mesh with $c_c = 4.0$.

Now, we ask ourselves how the tuning of the parameter τ_c affects when finer meshes are used. In particular, we want to know how important the c_c parameter becomes when higher order FEs are used. To answer this question we solve the TGV problem for two different discretizations: 32^3 $Q2/Q1$ and 21^3 $Q3/Q2$ elements meshes, and three different values of c_c: 0.0, 1.0 and 4.0.

Fig. 11 depicts the total energy dissipation rate (Fig. 11(a)) and the velocity divergence $L2$-norm (Fig. 11(b)) for the two discretizations considered and for different choices of c_c. We still see a dependence on c_c, but differences reduce when higher order of interpolation is used, as expected. It is also seen that

17

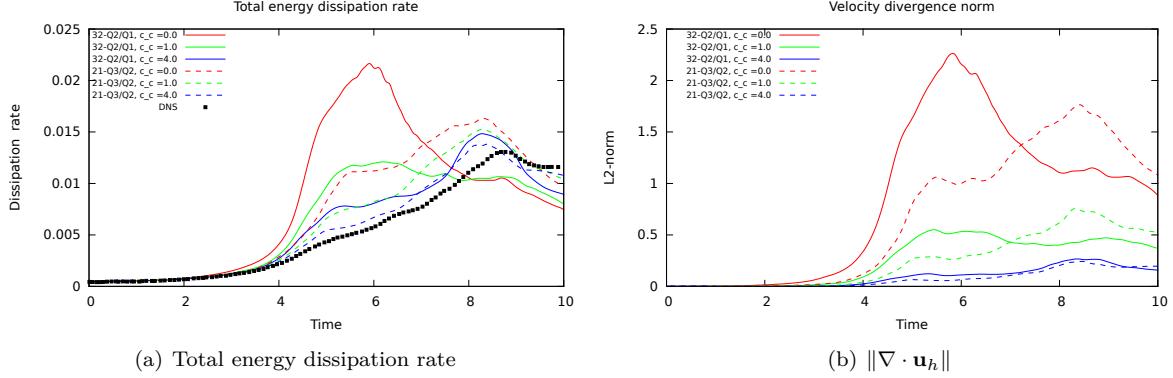(a) Total energy dissipation rate   (b) $\|\nabla \cdot \mathbf{u}_h\|$

Figure 11: Total energy dissipation and velocity divergence $L2$-norm for different c_c choices for OSS-ISS with 64^3 velocity DOFs

when increasing c_c, the differences between the two discretizations are reduced (for a fixed c_c). This behaviour is particularly significant when we look at the velocity divergence norm in Fig. 11(b), where we see that for $c_c = 0$ the two discretizations give a completely different evolution of $\|\nabla \cdot \mathbf{u}_h\|$, while for $c_c = 4.0$ the results are almost the same. In the same figure, we can see that the largest value of $\|\nabla \cdot \mathbf{u}_h\|$ for c_c is larger than the largest one depicted in Fig. 9(b), which is a result of a 16^3 $Q2/Q1$ elements mesh.

Going further we also do the same test for the 48^3 $Q2/Q1$ and 32^3 $Q3/Q2$ elements mesh (Fig. 12). In this figure we see that the differences between the three c_c cases are reduced. Looking at Fig. 12(a) it is seen that for $c_c = 0$ and $Q2/Q1$ elements, although the result is far from the DNS, the maximum value of the dissipation rate is much lower than the given in Fig. 11(a). When using $Q3/Q2$ elements, the changes on c_c produce lower differences compared against the $Q2/Q1$ approximation. The divergence norm depicted in Fig. 12 also show improvements with respect to Fig. 11. In this case, the maximum value of the divergence for c_c for $Q2/Q1$ elements is lower than the case of 64^3 velocity DOFs and also for the case of 32^3 velocity DOFs.

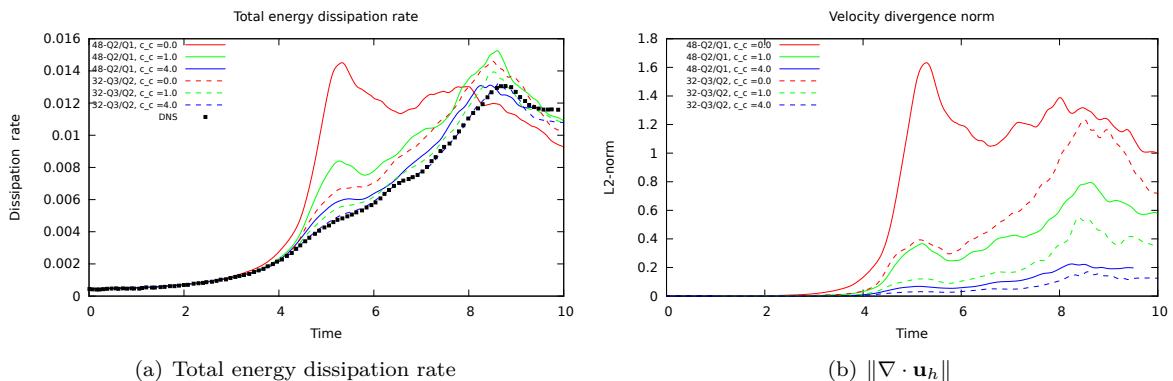(a) Total energy dissipation rate   (b) $\|\nabla \cdot \mathbf{u}_h\|$

Figure 12: Total energy dissipation and velocity divergence $L2$-norm for different c_c choices for OSS-ISS with 96^3 velocity DOFs

To summarize, we have checked that refining the mesh we converge to the DNS results and, for a given number of DOFs, the results improve when higher order interpolation is used. On the hand, we have seen that increasing c_c we reduce $\|\nabla \cdot \mathbf{u}\|$, and consequently the accuracy of the solution.

5.3. Turbulent channel flow at $Re_\tau = 395$

In Sect. 5.2 we have tested an homogeneous turbulent flow. Now, we want to check the behaviour of the proposed OSS-ISS stabilization method for a wall bounded turbulent test. To do so, we use the

Turbulent Channel Flow (TCF) test with a Reynolds number based on the wall friction (Re_τ) equal to 395. This benchmark was exhaustively tested using a VMS method with OSS in [16].

5.3.1. Test setting

The domain of the TCF problem for $Re_\tau = 395$ is given by a box of length ($2\pi\delta \times 2\delta \times 2/3\pi\delta$). 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 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. 13 depicts schematically the geometry of the computational domain for this test.
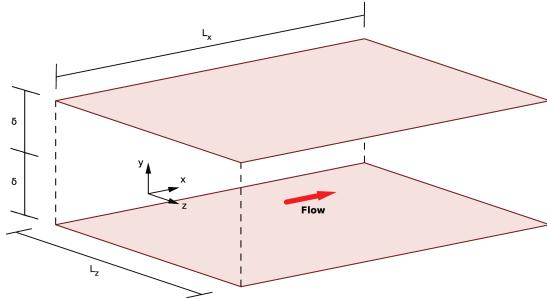

Figure 13: Channel computational domain.

The problem is solved using two different meshes, with 32^3 $Q2 - Q1$ and 21^3 $Q3/Q2$ elements mesh. Both meshes have refined elements near the wall in the wall-normal direction, like the one used in [16]. The algorithmic constants that appear in the OSS-ISS method will be discussed in the following subsections.

The obtained results are compared against a DNS computed in [42, 43] (MKM-DNS), then, the parameter election will be according to the ones defined in the cited paper. The bulk mean velocity and the half channel height are taken equal to one, $\bar{U} = 1$ and $\delta = 1$. Knowing the estimated Reynolds number based on the bulk mean velocity, $Re = \bar{U}2\delta/\nu \approx 13,750$ (see [44]), one can obtain the value of the viscosity, $\nu = 1.4545 \cdot 10^{-4}$. From the Reynolds number based on the friction velocity, we can determine the friction velocity magnitude: $u_\tau = Re_\tau \nu/\delta = 5.745 \cdot 10^{-2}$. Thus, the wall shear stress reads $\tau_w = u_\tau^2 = 3.3010 \cdot 10^{-3}$. A force equivalent to a pressure gradient is imposed to drive the movement of the flow in the stream-wise direction, $f_x = \tau_w/\delta$.

In order to achieve the statistically steady state solution, an initial solution is provided following [45]. This initial solution consists in a unidirectional velocity profile over which is added a fluctuation:

$$u_x = C\left(1 - y^8\right) + \epsilon \frac{L_x}{2} \sin(\pi y) \cos\left(\frac{4\pi x}{L_x}\right) \sin\left(\frac{2\pi z}{L_z}\right), \quad (47)$$

$$u_y = -\epsilon(1 + \cos(\pi y)) \sin(\pi y) \sin\left(\frac{4\pi x}{L_x}\right) \sin\left(\frac{2\pi z}{L_z}\right),$$

$$u_z = -\epsilon \frac{L_z}{2} \sin\left(\frac{4\pi x}{L_x}\right) \sin(\pi y) \cos\left(\frac{2\pi z}{L_z}\right).$$

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

5.3.2. Effect of the pressure subscale term on the conservation of mass

As it has been said in Sect. 3.4, the pressure subscale term has a noticeable effect on the solution when ISS elements are used. The effect of this term has also been analyzed in a previous test, see Sect. 5.2. Here we will also assess the effect of this term on a wall-bounded flow.

First we will focus on the effect of the second term in (20), looking how the c_2/c_1 ratio affects the solution keeping $c_c = 1.0$. As we are in a turbulent regime, we do not expect that the viscous counterpart

in (18) will have relevance on the solution, but we do expect it for the convective counterpart. Then, we will keep $c_1 = 12.0$ and we will increase c_2 from 1.0 to 16.0. The variations on c_2 not only have an effect on τ_c but also on τ_m. We solve the problem from $t = 0$ to $t = 20\pi$ (time needed to cross the channel 10 times, based on the initial mean bulk velocity $\bar{U}$) starting from the initial solution (47). The energy and $\|\nabla \cdot \mathbf{u}_h\|$ evolution are plotted in Fig. 14.

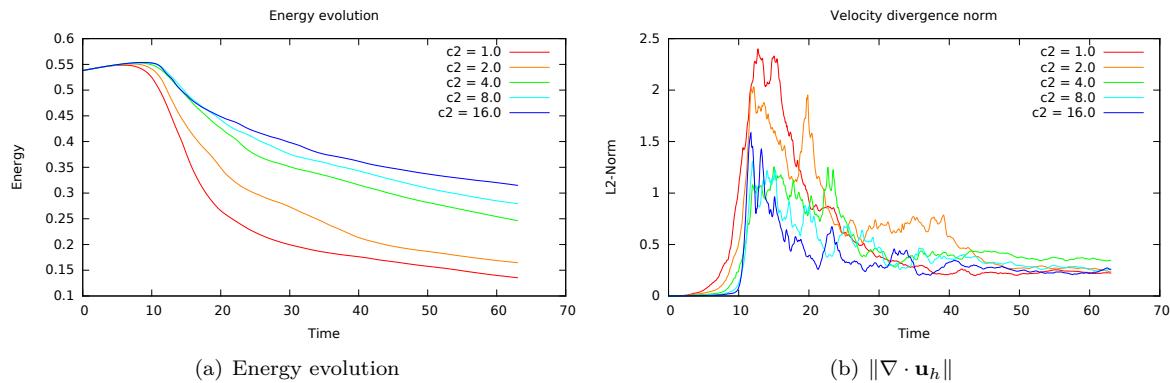(a) Energy evolution

(b) $\|\nabla \cdot \mathbf{u}_h\|$

Figure 14: Energy evolution and velocity divergence norm for different values of c_2, keeping $c_1 = 12.0$ and $c_c = 1.0$

It is clearly seen in Fig. 14 that the modification of c_2 has an effect on the solution. We see in Fig. 14(a) that the energy drops faster when lower values of c_2 are used. Looking at Fig. 14(b) we see that the velocity divergence L_2-norm is higher when lower values of c_2 are taken. The evolution of $\|\nabla \cdot \mathbf{u}_h\|$ gives us information about how the flow is evolving. From an initial and structured condition, the flow starts becoming chaotic around $t = 10$, depending on the case, when the turbulent structures increase the rotation of the flow particles, growing the spatial derivatives. At this stage, the energy dissipates until the equilibrium between the internal energy and the external forces is reached. These stages can be seen in Fig. 15, where we depict the vorticity isosurfaces for $|\omega| = 5.0$ coloured with the velocity field module at $t = 0.15$ (Fig. 15(a)), at $t = 12.0$ (Fig. 15(b)) and at $t = 70.0$ (Fig. 15(d)) setting $c2 = 1.0$ and $c_c = 32.0$, and using a 32^3 $Q2/Q1$ elements mesh.

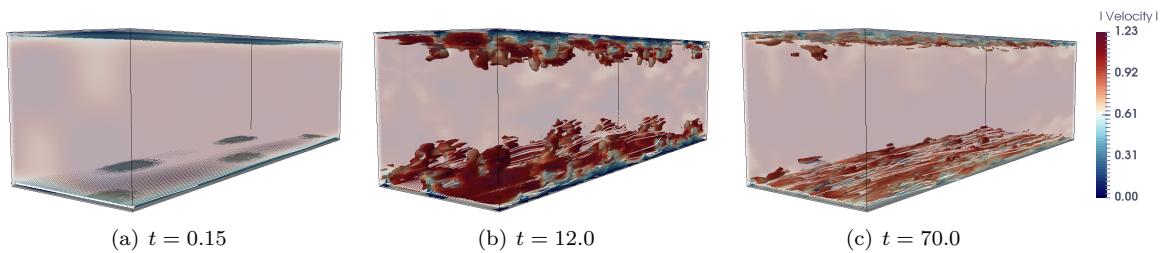
(a) $t = 0.15$

(b) $t = 12.0$

(c) $t = 70.0$

Figure 15: Velocity module at different times with $c_1 = 12.0$, $c_2 = 1.0$ and $c_c = 32.0$ using a 32^3 $Q2/Q1$ elements mesh.

When increasing c_2 we are decreasing the value of τ_m, but increasing τ_c. This means that in the energy dissipation equation (34), the dissipation through the convective term $\|\tau_m^{1/2}(\mathbf{u}_h \cdot \nabla \mathbf{u}_h - \boldsymbol{\eta}_h)\|^2$ becomes less relevant in front of the $\|\tau_c^{1/2}\nabla \cdot \mathbf{u}_h\|^2$.

Now, knowing the influence of c_2, and with the aim to distinguish the effect of modifying τ_m or τ_c, we keep this constant fixed with a value $c_2 = 8.0$ and we analyze the influence of τ_c on the solution, by changing c_c. In this situation, τ_m will remain constant for all cases and we will see the effect of the pressure subscale term. For this test, we solve the problem until $t = 300$, where the energy evolution stabilizes for all c_c choices. In Fig. 16 we see that the energy evolution (Fig. 16(a)) and the divergence norm evolution (Fig. 16(b)) follow the same pattern observed in Fig. 14. In this case we see that the improvement on the mass conservation is more pronounced when increasing c_c than when increasing c_2.

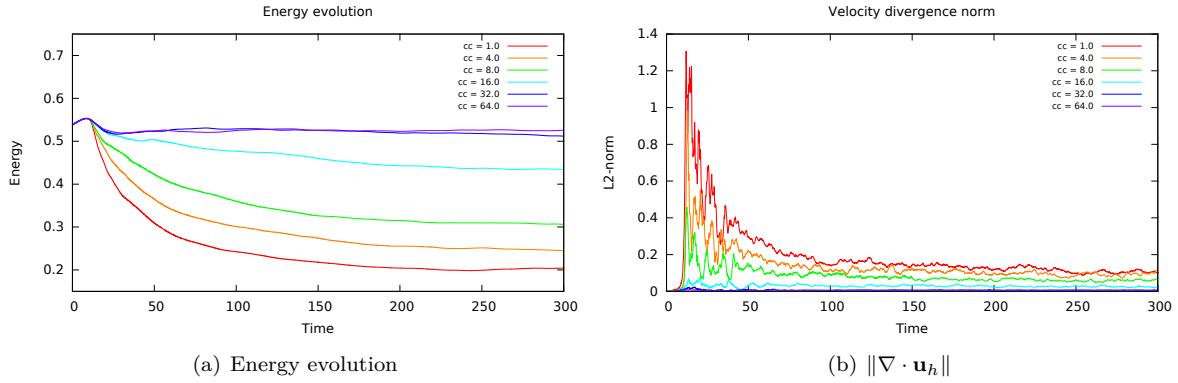(a) Energy evolution   (b) $\|\nabla \cdot \mathbf{u}_h\|$

Figure 16: Energy evolution and velocity divergence norm for different values of c_c, keeping $c_1 = 12.0$ and $c_2 = 8.0$

We also see in Fig. 16(a) that there is a certain threshold after which the result is improving very little. In this case, the results for $c_c = 32.0$ and $c_c = 64.0$ are almost the same.

We can compare our results with the MKM-DNS, contrasting the mean velocity and its fluctuations in Fig. 17. To obtain these results we have solved the problem from $t = 300$ to $t = 330$ with a time step size of $\delta t = 0.03$, collecting 1000 samples to obtain the mean quantities. The mean quantities are computed integrating the desired quantity within each element and adding up all elemental results belonging to the same y-orthogonal plane. Thus, since we are using high-order elements, the number of points that will appear in the graphics will be smaller than the number of DOFs. This procedure was also followed in [46] noting that the integral averaging procedure takes into account information from the secondary nodes that the widely used point-wise averaging does not contemplate.

Fig. 17(a) depicts the mean stream-wise velocity normalized by the prescribed wall-shear velocity, u_τ. In this picture we see the difference when changing c_c, where the dissipation of energy shown in Fig. 16(a) becomes clear looking at the velocity magnitude. The increase of $\|\nabla \cdot \mathbf{u}_h\|$, and its consequent loss of mass conservation, results in a mean velocity profile much lower than the DNS. We see that for $c_c = 32.0$ and $c_c = 64.0$ the mean velocity profile is on top of the DNS result, even for the coarse mesh in which we are solving the problem. Looking at the stream-wise velocity fluctuation (Fig. 17(b)) we see that it increases as we increase c_c and, in fact, the maximum becomes larger than the one obtained by DNS. This behaviour is justified by the fact that for low c_c values we have less energy in the system, and then, the fluctuating magnitude is also lower. The wall-normal velocity fluctuations are underpredicted for all the values of c_c the closer ones being obtained with $c_c = 4$ and $c_c = 8$ and diminishing for higher values. The span-wise velocity fluctuations are over-predicted (except for $c_c = 1$) but they become closer to DNS as c_c increases.

From Fig. 17 we see that increasing c_c the results for the mean values clearly improve, but this procedure may have some drawbacks. One of them is the ill-conditioning of the system of equations when $c_c \to \infty$. Table 1 summarizes the total accumulated solver iterations as well as the elapsed time needed to solve the problem for the different choices of c_c. As we see in Table 1, to solve the TCF

Table 1: Solver iterations and elapsed time to solve the TCF problem from $t = 300$ to $t = 330$ for different c_c values.

c_c	Solver iterations	Elapsed time (s)	Increment in time (%)
1.0	397899	19238	0.0
4.0	402490	19674	2.27
8.0	422217	19987	3.89
16.0	466687	20803	8.13
32.0	539879	22794	18.48
64.0	626487	24817	29.00

problem with $c_c = 64.0$ is 29.0% more expensive than solving it with $c_c = 1.0$. Thus, it is clear that at

(a) Mean streamwise velocity

(b) Rms streamwise velocity fluctuation

(c) Rms wall-normal velocity fluctuation
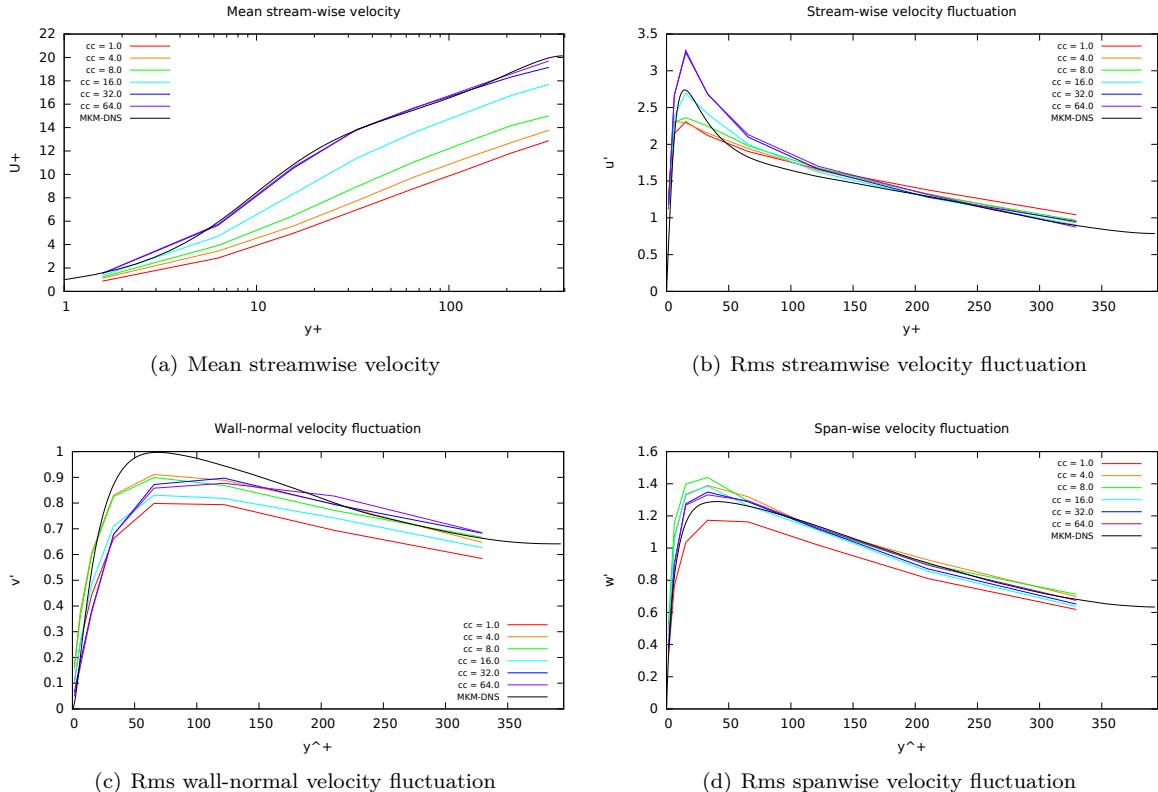
(d) Rms spanwise velocity fluctuation

Figure 17: Mean stream-wise velocity and rms velocity fluctuations using a 16^3 $Q2/Q1$ mesh for different choices of c_c.

some point it could be preferable to refine the mesh rather than increase the c_c value if we want more accurate results.

Provided that in all results shown until now, the case $c_c = 32.0$ gives almost the same solution as the $c_c = 64.0$ case, hereinafter we will only consider the case in which the parametric constants are $c_1 = 12.0$, $c_2 = 8.0$ and $c_c = 32.0$.

5.3.3. Refinement for a given c_c

Let us now explore what happens when the mesh is refined in the TCF test. Here we consider four different discretizations changing both the element size and the order of interpolation: 16^3 $Q2/Q1$, 24^3 $Q2/Q1$, 16^3 $Q3/Q2$ and 32^3 $Q2/Q1$. These discretizations have 32^3, 48^3, 48^3 and 64^3 velocity DOFs, respectively.

Looking at the energy evolution (Fig. 18(a)) and the velocity divergence L_2-norm (Fig. 18(b)) depicted in Fig. 18 we see that there are very little differences between the cases considered in this section. This means that given the appropriate parametric constants, the energy evolves in a similar way for all discretizations, with a similar evolution of the mass conservation.

If we focus on the averaged turbulent quantities shown in Fig. 19, we also see similar results between the different cases. In Fig. 19(a) the mean stream-wise velocity is plotted, and it is seen that all methods are almost on top of the DNS curve. The 16^3 $Q3/Q2$ discretization gives a mean velocity profile slightly lower than the other discretizations, which is also reflected on the energy evolution shown in Fig. 18(a). If we look at the velocity fluctuations in span-wise, wall-normal, and span-wise directions depicted in Fig. 19(b), Fig. 19(c) and Fig. 19(d), respectively, the convergence to the DNS solution is clear.

5.4. Turbulent channel flow at $Re_\tau = 590$

In order to assess the behaviour of the OSS-ISS method when the problem parameters change, we test the TCF at $Re_\tau = 590$. We compare our results against the DNS computations provided in [42]
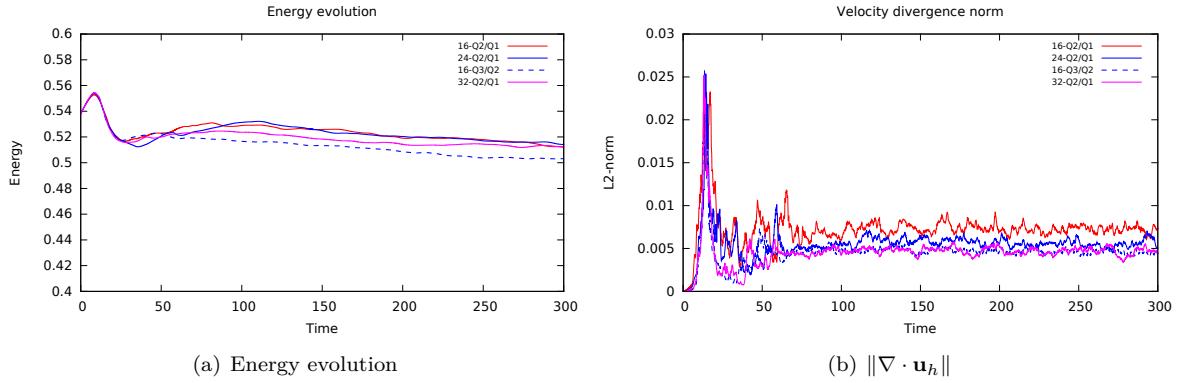
(a) Energy evolution

(b) $\|\nabla \cdot \mathbf{u}_h\|$

Figure 18: Energy evolution and velocity divergence norm refining the mesh, keeping $c_1 = 12.0$, $c_2 = 8.0$ and $c_c = 32.0$

(a) Mean stream-wise velocity

(b) Rms stream-wise velocity fluctuation

(c) Rms wall-normal velocity fluctuation
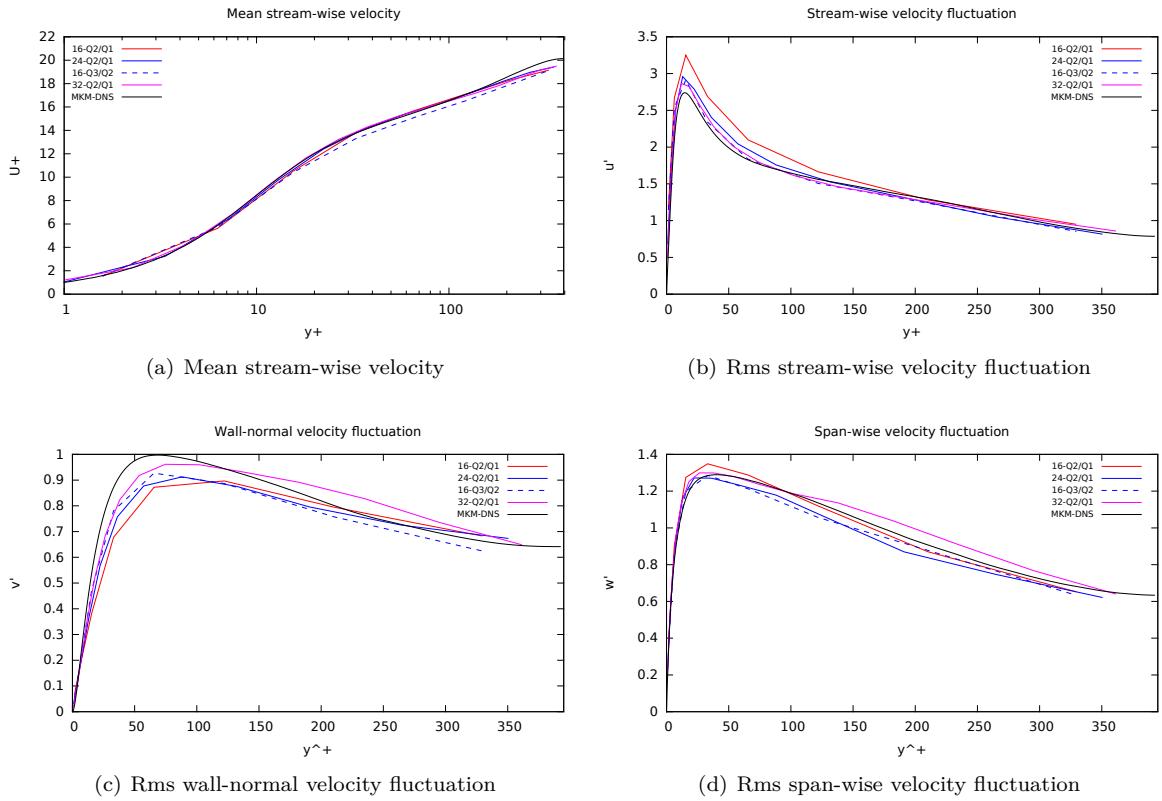
(d) Rms span-wise velocity fluctuation

Figure 19: Mean stream-wise velocity and rms velocity fluctuations for using different discretizations.

(MKM-DNS). Moreover, we also compare against other results obtained with similar methods such the one obtained by Gamnitzer et al in [46] and those obtained by Masud et al. in [47]. These two last works also use a VMS approach as an ILES method for the simulation of turbulent incompressible flows. Particularly, in [46], Gamnitzer et al. use a VMS method with both quasi-static and dynamic definition of the velocity subscale. We will compare against the results obtained with the quasi-static version, although the dynamic definition give very similar results for this test. Regarding the work by Masud et al. (see [47]), they use a VMS approach with time-dependent bubble functions for the subscales. In this work, they also evaluate the effect of the pressure subscale (the grad-div term) on the results.

23

5.4.1. Test setting

In this case, following the works [46, 47], we define a computational domain given by a box of length $(2\pi\delta \times 2\delta \times \pi\delta)$, with $\delta = 1$. Homogeneous Dirichlet boundary conditions for the velocity DOFs are also 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. The time step size is set equal to $\delta t = 0.0015$. We let the flow evolve until the solution is statistically stable, then we collect the statistics.

For this test we use a 21^3 $Q3/Q2$ elements mesh. Note that in [46, 47] they use a 64^3 $Q1/Q1$ elements mesh, which has almost the same velocity DOFs, but more pressure DOFs. The mesh is stretched near the wall in the wall-normal direction, following the rule used in [16], with a stretching parameter $\gamma = 2.5$. The algorithmic constants that appear in the OSS-ISS method are the ones that has given best results in the previous test: $c_1 = 12.0$, $c_2 = 8.0$ and $c_c = 32.0$.

5.4.2. Results

In Fig. 20 we depict the averaged profiles of the mean stream-wise velocity (Fig.20(a)), and the velocity fluctuations in stream-wise, wall-normal and span-wise components (Fig. 20(b), Fig. 20(c) and Fig. 20(d), respectively). Looking at the mean stream-wise velocity profile shown in Fig. 20(a), we see
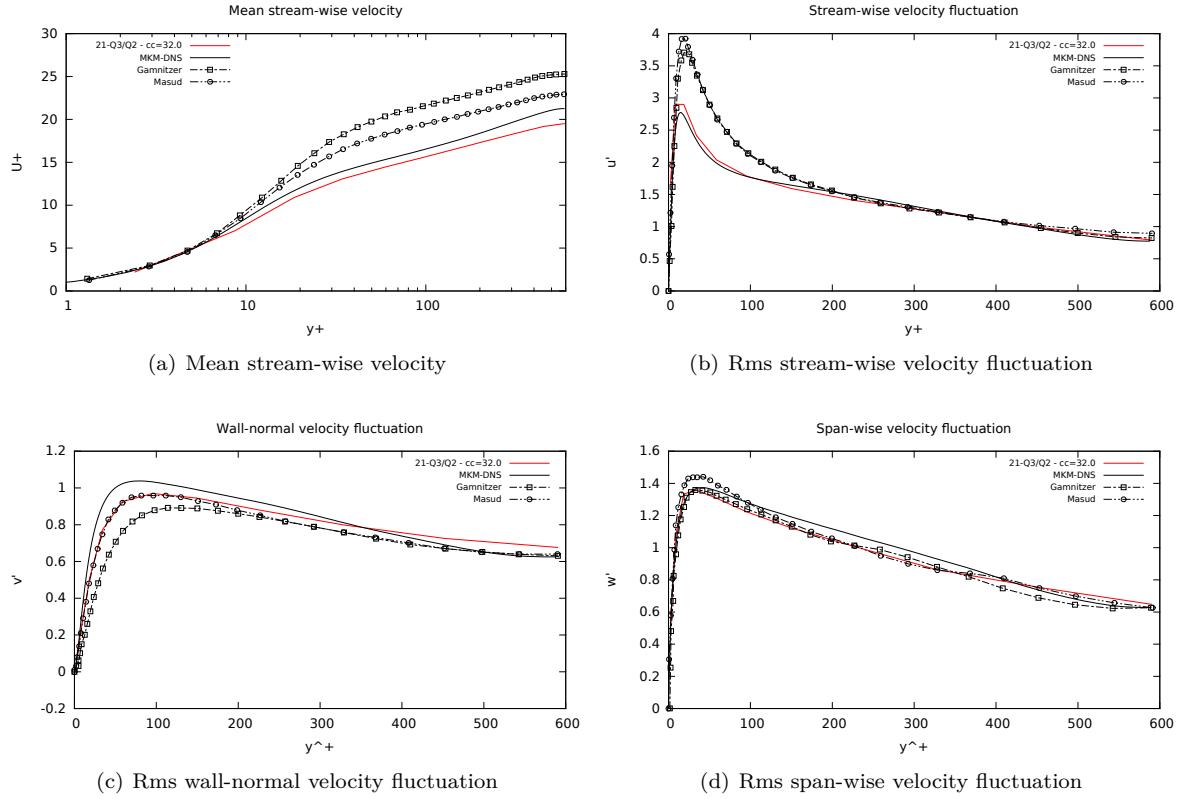

(a) Mean stream-wise velocity

(b) Rms stream-wise velocity fluctuation

(c) Rms wall-normal velocity fluctuation

(d) Rms span-wise velocity fluctuation

Figure 20: Mean stream-wise velocity and rms velocity fluctuations for TCF test with $Re_\tau = 590$.

that the results obtained with the OSS-ISS method are in quite good agreement with the DNS, although the computed profile is slightly under the expected. Moreover, comparing against the ones obtained in [46] and [47], we see that we get an improvement on the solution of the mean velocity profile.

We also see a very good agreement with the DNS in the stream-wise velocity fluctuations, shown in Fig. 20(b). In this figure, the improvement with respect to the results given in [46] and [47] is also revealed, especially in the near-wall points. Similar results are observed in the wall-normal and span-wise velocity fluctuations, shown in Fig. 20(c) and Fig. 20(d), respectively, where our solution is very close to the cited results near the wall, but slightly over-predicted at the furthest points.

Regarding the effect of the grad-div stabilization term, it is shown in [47] that there is an improvement on the velocity divergence norm when this term is considered, a result that we also have noticed in this work. In the cited results, when the pressure subscale term is added, the time-averaged L_2-norm of the velocity divergence normalized by volume of the domain is $< \|\nabla \cdot \mathbf{u}_h\| > /\|1\| = 16.7$. In our computations this quantity is $< \|\nabla \cdot \mathbf{u}_h\| > /\|1\| = 3.21$, noting that the incompressibility constraint is better satisfied.

6. Conclusions

In this article we have considered the numerical simulation of turbulent incompressible flows using the term-by-term OSS method for convection-stabilization applied to ISS elements. We have considered an implicit treatment of the projection. For the solution of the monolithic linear systems a block preconditioning strategy that makes use of recursive block factorizations has been proposed. Among the three variants we proposed, namely diagonal, lower and upper triangular, the last two are much faster than the first one, as expected, the upper triangular being slightly faster for the problems considered.

Using this strategy, the comparison of the three methods, ASGS, term-by-term OSS, and convection-only OSS with ISS elements reveals that the accuracy is similar for the same order of interpolation of the velocity, the OSS-ISS being slightly inferior in this respect. But on the other hand, when computational cost is analyzed the OSS-ISS is clearly the cheapest one so a finer discretization can be used. Another advantage of the OSS-ISS method is that it does not change the nature in time of the problem, i.e., the discrete system is an index-2 differential-algebraic equation. Thus, it allow us, e.g., to use Segregated Runge-Kutta (SRK) time integration schemes that cannot be applied to the ASGS or OSS methods. This issue will be discussed in further works, but we can say that SRK schemes are well suited to perform high-order time integration as well as the use of an adaptive time-stepping technique. This method was firstly introduced and tested for laminar incompressible flows in [39].

Finally, we have also analyzed the influence of the grad-div stabilization on the results when ISS discretizations are considered. It has been clearly shown that this term affects the solution. Increasing the value of c_c results in a better mass conservation and better results, but this saturates at some point, from which little improvements are achieved while a high increase on the computational cost is observed. This effect is smaller when the mesh is refined and also when higher order interpolations are used. A word of caution is necessary here, as the theoretical result on the convergence of the TH solution to the SV one when $\tau_c \to \infty$ is only valid on a mesh that is a barycentre-refinement of a regular mesh. In our numerical experiments we have not used this type of meshes and the solution actually improves when increasing c_c but this is not the case in, e.g., the driven cavity on a uniform tetrahedral mesh [31]. Moreover, we have tested the behaviour of the proposed methods when the problem parameters change, i.e., changing the Reynolds number. In that case we have observed that the results remain accurate for a fixed value of c_c.

Acknowledgements

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] A. N. Brooks, T. J. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering 32 (1-3) (1982) 199–259.

[6] 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.

[7] M. Olshanskii, G. Lube, T. Heister, J. Löwe, Grad-div stabilization and subgrid pressure models for the incompressible Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering 198 (49-52) (2009) 3975–3988.

[8] 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.

[9] 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.

[10] 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.

[11] B. Koobus, C. Farhat, A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes–application to vortex shedding, Computer Methods in Applied Mechanics and Engineering 193 (15-16) (2004) 1367–1383.

[12] V. John, A. Kindl, Variants of projection-based finite element variational multiscale methods for the simulation of turbulent flows, International Journal for Numerical Methods in Fluids 56 (8) (2008) 1321–1328.

[13] V. John, A. Kindl, Numerical studies of finite element variational multiscale methods for turbulent flow simulations, Computer Methods in Applied Mechanics and Engineering 199 (13-16) (2010) 841–852.

[14] R. Calderer, A. Masud, Residual-based variational multiscale turbulence models for unstructured tetrahedral meshes, Computer Methods in Applied Mechanics and Engineering 254 (2013) 238–253.

[15] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Computer Methods in Applied Mechanics and Engineering 197 (1-4) (2007) 173–201.

[16] 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.

[17] 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.

[18] R. Codina, J. Principe, O. Guasch, S. Badia, Time dependent subscales in the stabilized finite element approximation of incompressible flow problems, Computer Methods in Applied Mechanics and Engineering 196 (21-24) (2007) 2413–2430.

[19] R. Codina, Analysis of a stabilized finite element approximation of the Oseen equations using orthogonal subscales, Applied Numerical Mathematics 58 (3) (2008) 264–283.

[20] 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.

[21] 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.

[22] M. A. Olshanskii, A low order Galerkin finite element method for the Navier–Stokes equations of steady incompressible flow: a stabilization issue and iterative methods, Computer Methods in Applied Mechanics and Engineering 191 (47–48) (2002) 5515–5536.

[23] G Lube, G Rapin, J Löwe, Local projection stabilization for incompressible flows: Equal-order vs. inf-sup stable interpolation, Electronic Transactions on Numerical Analysis 32 (2008) 106–122.

[24] M. Olshanskii, A. Reusken, Grad-div stablilization for Stokes equations, Mathematics of Computation 73 (248) (2004) 1699–1718.

[25] 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.

[26] 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.

[27] N. Ahmed, T. C. Rebollo, V. John, S. Rubino, A Review of Variational Multiscale Methods for the Simulation of Turbulent Incompressible Flows, Archives of Computational Methods in Engineering (2015) 1–50.

[28] L. P. Franca, T. J. R. Hughes, Two classes of mixed finite element methods, Computer Methods in Applied Mechanics and Engineering 69 (1) (1988) 89–129.

[29] T. Gelhard, G. Lube, M. A. Olshanskii, J.-H. Starcke, Stabilized finite element schemes with LBB-stable elements for incompressible flows, Journal of Computational and Applied Mathematics 177 (2) (2005) 243–267.

[30] E. W. Jenkins, V. John, A. Linke, L. G. Rebholz, On the parameter choice in grad-div stabilization for the Stokes equations, Advances in Computational Mathematics 40 (2) (2013) 491–516.

[31] M. Case, V. Ervin, A. Linke, L. Rebholz, A connection between Scott-Vogelius and grad-div stabilized Taylor-Hood FE approximations of the Navier-Stokes equations, SIAM Journal on Numerical Analysis 49 (4) (2011) 1461–1481.

[32] L. R. Scott, M. Vogelius, Conforming finite element methods for incompressible and nearly incompressible continua, Lectures in Applied Mathematics 22 (1985) 221–244.

[33] J. Principe, R. Codina, F. Henke, The dissipative structure of variational multiscale methods for incompressible flows, Computer Methods in Applied Mechanics and Engineering 199 (13–16) (2010) 791–801.

[34] A. Linke, L. G. Rebholz, N. E. Wilson, On the convergence rate of grad-div stabilized Taylor-Hood to Scott-Vogelius solutions for incompressible flow problems, Journal of Mathematical Analysis and Applications 381 (2) (2011) 612–626.

[35] S. Zhang, A family of $Q_{k+1,k} \times Q_{k,k+1}$ divergence-free finite elements on rectangular grids, SIAM Journal on Numerical Analysis 47 (3) (2009) 2090–2107.

[36] M. Yano, D. L. Darmofal, BDDC preconditioning for high-order Galerkin Least-Squares methods using inexact solvers, Computer Methods in Applied Mechanics and Engineering 199 (45–48) (2010) 2958–2969.

[37] 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.

[38] 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.

[39] O. Colomés, S. Badia, Segregated Runge–Kutta methods for the incompressible Navier–Stokes equations, International Journal for Numerical Methods in Engineering 105 (5) (2016) 372–400.

[40] M. Brachet, Direct simulation of three-dimensional turbulence in the Taylor—Green vortex, Fluid Dynamics Research 8 (1-4) (1991) 1–8.

[41] G. J. Gassner, A. D. Beck, On the accuracy of high-order discretizations for underresolved turbulence simulations, Theoretical and Computational Fluid Dynamics 27 (3-4) (2013) 221–237.

[42] R. D. Moser, J. Kim, N. N. Mansour, Direct numerical simulation of turbulent channel flow up to Re_τ=590, Physics of Fluids 11 (4) (1999) 943–945.

[43] 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.

[44] S. B. Pope, Turbulent Flows, Cambridge University Press, 2000.

[45] 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.

[46] P. Gamnitzer, V. Gravemeier, W. A. Wall, Time-dependent subgrid scales in residual-based large eddy simulation of turbulent channel flow, Computer Methods in Applied Mechanics and Engineering 199 (13-16) (2010) 819–827.

[47] A. Masud, R. Calderer, A variational multiscale method for incompressible turbulent flows: Bubble functions and fine scale fields, Computer Methods in Applied Mechanics and Engineering 200 (33-36) (2011) 2577–2593.