Computers & Fluids 117 (2015) 168–182

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Semi-implicit BDF time discretization of the Navier–Stokes equations with VMS-LES modeling in a High Performance Computing framework

Davide Forti, Luca Dedè *

CMCS – Chair of Modeling and Scientific Computing, MATHICSE – Mathematics Institute of Computational Science and Engineering, EPFL – École Polytechnique Fédérale de Lausanne, Station 8, Lausanne CH-1015, Switzerland

ARTICLE INFO

Article history:
Received 3 April 2015
Received in revised form 13 May 2015
Accepted 14 May 2015
Available online 23 May 2015

Keywords:
Navier–Stokes equations
Variational Multiscale modeling
Backward Differentiation Formulas
Finite Element method
Large Eddy Simulation
High Performance Computing

ABSTRACT

In this paper, we propose a semi-implicit approach for the time discretization of the Navier–Stokes equations with Variational Multiscale-Large Eddy Simulation turbulence modeling (VMS-LES). For the spatial approximation of the problem, we use the Finite Element method, while we employ the Backward Differentiation Formulas (BDF) for the time discretization. We treat the nonlinear terms arising in the variational formulation of the problem with a semi-implicit approach leading to a linear system associated to the fully discrete problem which needs to be assembled and solved only once at each discrete time instance. We solve this linear system by means of the GMRES method by employing a multigrid (ML) right preconditioner for the parallel setting. We validate the proposed fully discrete scheme towards the benchmark problem of the flow past a squared cylinder at high Reynolds number and we show the computational efficiency and scalability results of the solver in a High Performance Computing framework.

1. Introduction

In this work, we consider the Navier–Stokes equations with Variational Multiscale-Large Eddy Simulation (VMS-LES) modeling of turbulence for incompressible Newtonian fluids at high Reynolds number, with the focus being on their efficient time discretization and, more in general, their numerical solution in a parallel setting. Indeed, the accurate and efficient simulation of turbulent flows still represents a major challenge, even in a High Performance Computing framework. The Direct Numerical Simulation (DNS) of turbulent flows requires the full representation of the whole range of spatial and temporal turbulent scales at the discrete level [24,38,40]; as consequence, even for moderate Reynolds numbers, the DNS may be unfeasible for several problems of practical interest. Conversely, with Large Eddy Simulation (LES) approaches [39,40,47] only the "large" scales of the flow field are fully represented and resolved at the discrete level, while the effect of the "small" unresolved scales is taken into account by means of suitable models based on the resolved scales, thus making the computational costs more affordable; among the others, examples of established LES models are the Smagorinsky [49] and dynamic Smagorinsky [18,35] models.

The concept of defining LES models for the Navier–Stokes equations by means of the Variational Multiscale method (VMS) ([29,34]) has been introduced in [31] and further extended in [32,33]; since then, the so called VMS-LES method has been widely developed and used for the numerical simulation of turbulent flows in several benchmarking and applicative contexts [1–4,8,13,20,25,28,30]. Indeed, the VMS-LES method is advantageous since it provides a unified framework for the definition of spatial approximation schemes which are stable in the sense of the well-posedness, capable of controlling the numerical instabilities arising in the convective regimes at high Reynolds numbers, and adequate to represent the turbulence LES modeling. In this framework, the LES model or filter stems from a low order approximation of the nonlinear terms deriving from the residual based VMS formulation applied to the incompressible Navier–Stokes equations.

In this work, we focus on efficient spatial and, above all, time discretization schemes for the incompressible Navier–Stokes equations with VMS-LES modeling of turbulence by aiming at the simulation of flows at high Reynolds number in a parallel framework. In particular, the problem is first discretized in space by using the Finite Element method [43], mainly with basis functions of low polynomial degree, and then in time by means of Finite Differences [41]. More specifically, we approximate the time derivative in the Navier–Stokes equations by means of Backward

* Corresponding author. Tel.: +41 21 6932909; fax: +41 21 6935510.
E-mail address: luca.dede@epfl.ch (L. Dedè).

http://dx.doi.org/10.1016/j.compfluid.2015.05.011
0045-7930/

Differentiation Formulas (BDF) of orders 1, 2, or higher; see e.g. [19,41]. Then, in order to obtain an efficient formulation, we propose a semi-implicit treatment of the nonlinear terms appearing in the fully discrete scheme, for which we "linearize" the discrete problem at each discrete time instance by means of temporal extrapolations based on Newton–Gregory backward polynomials [44]. The approach proposed leads to assemble and solve a unique linear system for each discrete time instance; conversely, a fully implicit approach, e.g. using the Newton method, would require, in principle, repeated assembly and solutions of a linear system. While our semi-implicit BDF method for VMS-LES modeling may require restrictions on the time step due to stability issues of the time discretization scheme, these can be offset by the larger computational costs associated instead to a fully implicit approach; in practice, our numerical experience indicates that stable solutions for the semi-implicit BDF scheme are obtained also for relatively "large" time steps, conversely to explicit schemes for which limitations on their size are generally more severe. We remark that, in literature, VMS-LES formulations of the Navier–Stokes equations are often discretized in time by means of implicit schemes [2,20,28] as the generalized-α method ([21]), low order semi-implicit schemes as the backward Euler method with Picard "linearization" [13], or explicit schemes as done in [25] with a Runge–Kutta scheme. In this respect, we remark that the main contribution of this work consists in defining a semi-implicit BDF scheme for the approximation of the Navier–Stokes equations with turbulence VMS-LES modeling, which allows efficient and accurate numerical solutions of the problem, without incurring in severe restrictions of the time step.

Since we aim at solving large scale problems at high Reynolds number, the use of parallel architectures is necessary [22]. In such parallel framework, in addition to suitable choices of the spatial and time discretization schemes, we stress the fact that the use of an efficient linear solver is often mandatory to make the numerical simulations computationally feasible. Nevertheless, the efficacy of the linear solver and preconditioning strategies for VMS-LES formulations are relatively little explored in literature; in [13] a balancing Neumann–Neumann domain decomposition method is used for preconditioning with the GMRES iterative solver, while in [20] algebraic multigrid ([46]) strategies for VMS-LES modeling are discussed. In this work, we combine our semi-implicit BDF time discretization scheme for the Navier–Stokes equations with VMS-LES modeling together with a parallel multigrid right preconditioner based on the package ML of *Trilinos* [17] applied to the GMRES linear iterative solver. In this manner, we obtain an efficient solver for the High Performance Computing implemented in the open-source Finite Element library *LifeV* [55].

The proposed numerical scheme is tested for the vortex shedding benchmark problem of an incompressible fluid flow at high Reynolds number ($Re = 22,000$) past a squared cylinder; this benchmark allows us to validate our numerical strategy against the results obtained with other LES models [25,45,50] and also by means of experimental tests [5,36]. In our study, we present and discuss the numerical results obtained by considering both first and second degree Lagrange polynomials for the Finite Element representation of the fluid velocity and pressure fields, first and second order semi-implicit BDF scheme, and different time steps. Furthermore, we analyze the energy spectra obtained for the spatial discretizations under consideration. Then, regarding the efficiency of the linear parallel solver, we investigate its strong scalability properties by solving the benchmark problem at hand on supercomputer machines. Finally, we discuss the choice of the stabilization parameters appearing in the VMS-LES formulation and their role in obtaining numerical simulations which feature affordable computational costs. In particular, we consider in this work stabilization parameters which involve the time step in the quasi-static approximation of the fine scale differential operator, similarly to [2]. In this respect, we compare, for the vortex shedding benchmark problem, the spatial distribution of stabilization parameters, chosen both as dependent and independent of the time step, and we highlight their impact on the efficiency of the GMRES solver with multigrid preconditioner.

The paper is organized as follows. In Section 2, after recalling the Navier–Stokes equations with VMS-LES modeling of turbulence, we introduce the spatial discretization of the coarse scale velocity and pressure fields by means of the Finite Element method. In Section 3, we focus on the semi-implicit time discretization of the problem by means of BDF formulas. Then, in Section 4, we describe the parallel solver developed for the fully discrete problem, as well as the multigrid preconditioner used. In Section 5, we report and discuss the numerical results on the squared cylinder benchmark problem and we analyze the choice of the stabilization parameters associated to the VMS-LES formulation. Conclusions follow in Section 6.

2. Variational Multiscale LES modeling for the Navier–Stokes equations

In this section we start by recalling the strong formulation of the incompressible Navier–Stokes, the VMS-LES modeling and the spatial approximation based on the Finite Element method.

2.1. The incompressible Navier–Stokes equations

Let $\Omega \subset \mathbb{R}^d$ be the spatial fluid domain with piecewise smooth boundary $\Gamma \equiv \partial\Omega$, for which we consider the dimension $d = 3$; we denote with Γ_D the subset of Γ where the essential (Dirichlet) boundary conditions are applied, while natural (Neumann) boundary conditions will be considered on $\Gamma_N = \Gamma \setminus \Gamma_D$. The Navier–Stokes equations for an incompressible fluid read:

$$\rho\frac{\partial \boldsymbol{u}}{\partial t} + \rho\boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nabla\cdot\boldsymbol{\sigma}(\boldsymbol{u},p) = \boldsymbol{f} \quad \text{in } \Omega\times(0,T), \tag{1}$$

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

$$\boldsymbol{u} = \mathbf{g} \quad \text{on } \Gamma_D\times(0,T), \tag{3}$$

$$\boldsymbol{\sigma}(\mathbf{u},p)\hat{\mathbf{n}} = \mathbf{h} \quad \text{on } \Gamma_N\times(0,T), \tag{4}$$

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

where the dependent variables $\boldsymbol{u}$ and p indicate the velocity and the pressure of the fluid, respectively, ρ is the density, $\boldsymbol{f}$ is the vector of external forces, $\hat{\mathbf{n}}$ is the outward directed unit normal vector to Γ_N, and $\boldsymbol{\sigma}$ is the stress tensor defined, for a Newtonian fluid, as:

$$\boldsymbol{\sigma}(\boldsymbol{u},p) = -p\mathbf{I} + 2\mu\boldsymbol{\epsilon}(\boldsymbol{u}); \tag{6}$$

we denoted by μ the dynamic viscosity of the fluid and by $\mathbf{I}$ the second order identity tensor, while $\boldsymbol{\epsilon}(\boldsymbol{u})$ is the strain tensor:

$$\boldsymbol{\epsilon}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right). \tag{7}$$

The functions $\mathbf{g}$ and $\mathbf{h}$ indicate the Dirichlet and Neumann data, respectively, while $\mathbf{u}_0$ the initial data. We recall that Eqs. (1)–(5) represent the balance of momentum in convective form, the mass conservation, the essential and natural boundary conditions, and the initial condition on the velocity, respectively.

In view of the spatial approximation of the Navier–Stokes equations, we introduce the infinite dimensional function spaces $\mathcal{V}_{\mathbf{g}} := \{\boldsymbol{u} \in \left[H^1(\Omega)\right]^d : \ \mathbf{u}|_{\Gamma_D} = \mathbf{g}\}$, $\mathcal{V}_{\mathbf{0}} := \{\boldsymbol{u} \in \left[H^1(\Omega)\right]^d : \ \mathbf{u}|_{\Gamma_D} = \mathbf{0}\}$, and $\mathcal{Q} := L^2(\Omega)$; in addition, we define the function spaces $\boldsymbol{\mathcal{V}}_{\mathbf{g}} := \mathcal{V}_{\mathbf{g}} \times \mathcal{Q}$ and $\boldsymbol{\mathcal{V}}_{\mathbf{0}} := \mathcal{V}_{\mathbf{0}} \times \mathcal{Q}$. Then, the weak formulation of the Navier–Stokes equations reads, for all $t \in (0,T)$ ([43]):

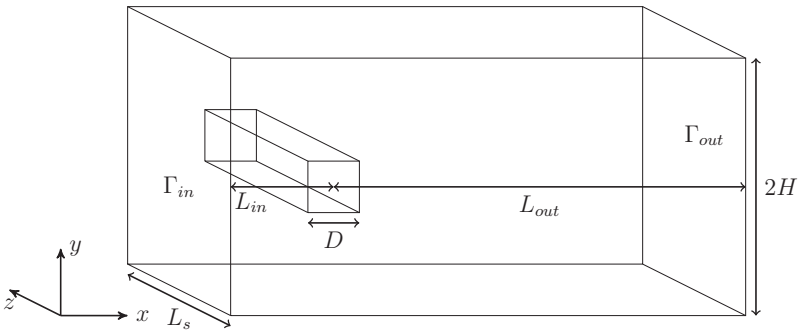


Fig. 1. Computational domain Ω considered for the squared cylinder benchmark problem.

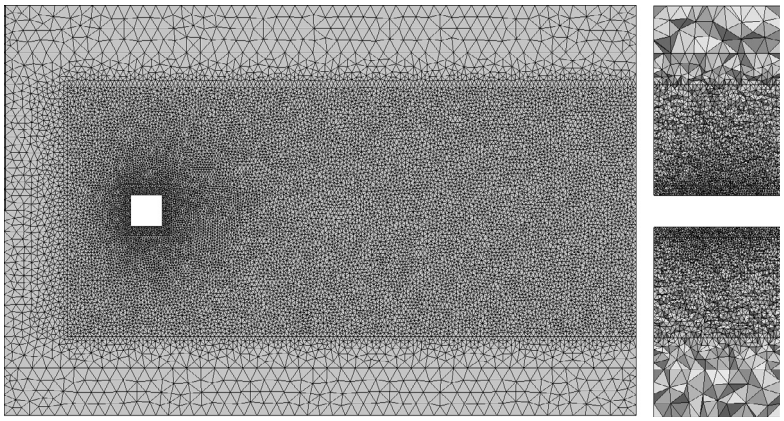


Fig. 2. Computational mesh used for the numerical simulation of the flow past a squared cylinder.

Table 1
Number of degrees of freedom used in our simulations for the mesh of Fig. 2.

FE discretization	Number of dofs
$\mathbb{P}1$–$\mathbb{P}1$	1,323,056
$\mathbb{P}2$–$\mathbb{P}2$	9,209,040

$$\begin{aligned}\text{find } \boldsymbol{U} = \boldsymbol{U}(t) = \{\boldsymbol{u}, p\} \in \boldsymbol{\mathcal{V}}_{\mathbf{g}} \\ : \left(\boldsymbol{w}, \rho \frac{\partial \boldsymbol{u}}{\partial t}\right) + (\boldsymbol{w}, \rho \boldsymbol{u} \cdot \nabla \boldsymbol{u}) + \left(\nabla \boldsymbol{w}, \mu\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right)\right) \\ - (\nabla \cdot \boldsymbol{w}, p) + (q, \nabla \cdot \boldsymbol{u}) \\ = (\boldsymbol{w}, \boldsymbol{f}) + (\boldsymbol{w}, \mathbf{h})_{\Gamma_N} \quad \text{for all } \boldsymbol{W} = \{\boldsymbol{w}, q\} \in \boldsymbol{\mathcal{V}}_{\mathbf{0}},\end{aligned} \tag{8}$$

where $\boldsymbol{u}(0) = \boldsymbol{u}_0$; $(\cdot,\cdot)$ denotes the standard L^2 inner product with respect to the spatial domain Ω and $(\cdot,\cdot)_{\Gamma_N}$ the one on Γ_N.

2.2. Spatial approximation: Finite Element method and VMS-LES modeling

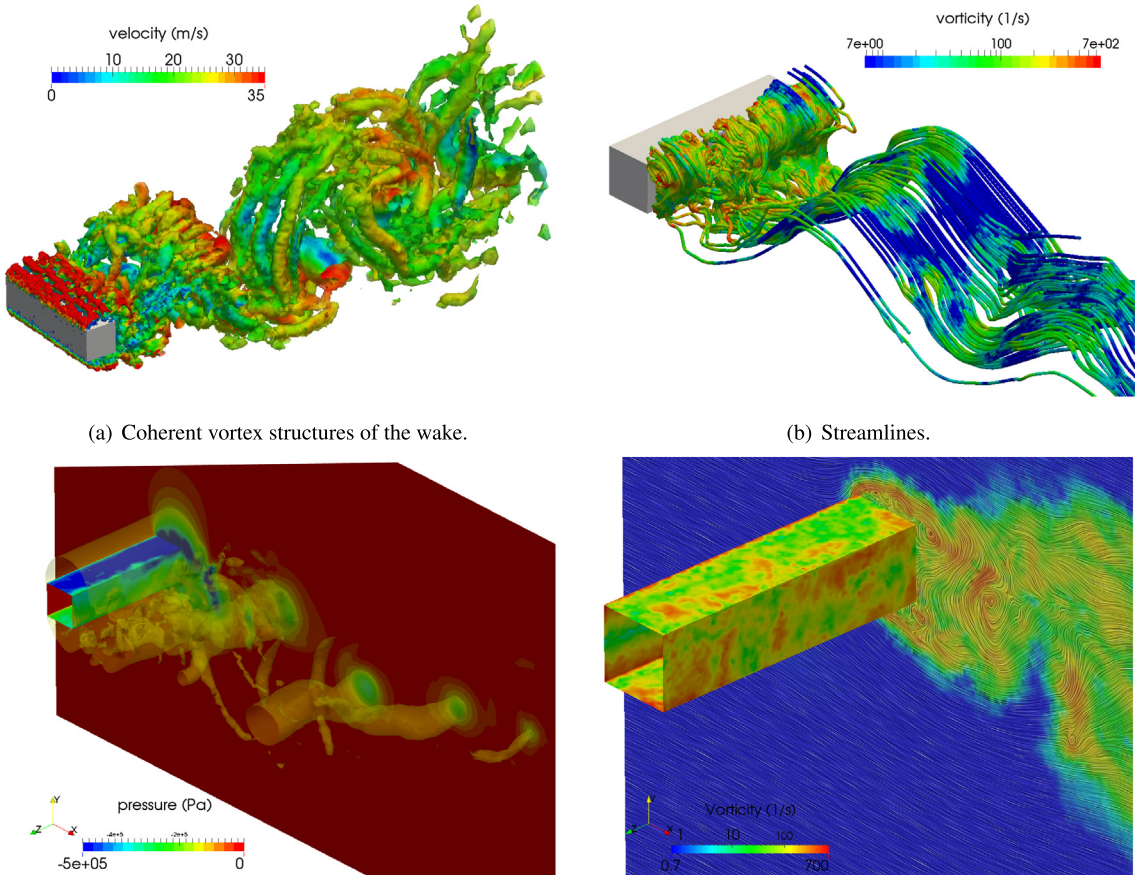
We consider the Variational Multiscale (VMS) [2,34] method applied to the standard weak form of the Navier–Stokes equations (8). The VMS method introduces a priori a decomposition of the solution into coarse and fine scales. Accordingly, the weak formulation of the Navier–Stokes equations (8) is split into coarse and fine scale subproblems. The coarse scale solution is identified with the numerical approximation given by the Finite Element method, while the fine scale component needs to be modeled. We remark that the fine scale component of the solution is often modeled analytically (in closed form), written in terms depending both on the data of the problem and the coarse scale solution, and finally substituted in the coarse scale subproblem. In virtue of a projection of the fine scale solution into the coarse one, a finite dimensional system for the coarse scale component of the solution is obtained.

Let us introduce at this point a suitable Finite Element discretization, specifically with piecewise Lagrange polynomials of degree $r \geqslant 1$ over the computational domain Ω that is triangulated with a mesh $\mathcal{T}_h$ comprised of tetrahedrons; in this respect, we indicate with $X_r^h := \left\{ v^h \in C^0(\overline{\Omega}) : v^h|_K \in \mathbb{P}_r, \text{ for all } K \in \mathcal{T}_h \right\}$ the function space of Finite Element and with h_K the diameter of the mesh element $K \in \mathcal{T}_h$.

We consider a multiscale direct-sum decomposition of a general function space $\boldsymbol{\mathcal{V}}$, which can be interpreted either as $\boldsymbol{\mathcal{V}}_{\mathbf{g}}$ or $\boldsymbol{\mathcal{V}}_{\mathbf{0}}$, into the coarse and fine scales subspaces as:

$$\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{V}}^h \oplus \boldsymbol{\mathcal{V}}', \tag{9}$$

where $\boldsymbol{\mathcal{V}}^h$ is the coarse scale function space associated to the Finite Element discretization and $\boldsymbol{\mathcal{V}}'$ is an infinite dimensional function space representing the fine scales not directly represented in the discretization; more specifically, we have $\boldsymbol{\mathcal{V}}^h = \mathcal{V}_{\mathbf{g}}^h \times \mathcal{Q}_h$ or $\boldsymbol{\mathcal{V}}^h = \mathcal{V}_{\mathbf{0}}^h \times \mathcal{Q}_h$, with $\mathcal{V}_{\mathbf{g}}^h := \mathcal{V}_{\mathbf{g}} \cap \left[X_r^h\right]^d$, $\mathcal{V}_{\mathbf{0}}^h := \mathcal{V}_{\mathbf{0}} \cap \left[X_r^h\right]^d$, and $\mathcal{Q}^h := \mathcal{Q} \cap X_r^h$. We remark that we use the same Finite Element space X_r^h of degree r to define the spatial discretization for both the velocity and pressure variables. Then, we have from Eq. (9) the following decompositions:



(a) Coherent vortex structures of the wake.

(b) Streamlines.

(c) Pressure field.

(d) Vorticity field.

Fig. 3. Post-processing of the results at time $t = 10$ s. Solution obtained using $\mathbb{P}2$–$\mathbb{P}2$ Finite Element, BDF2, and $\Delta t = 0.0025$ s.

$$\boldsymbol{w} = \boldsymbol{w}^h + \boldsymbol{w}', \tag{10}$$

$$q = q^h + q', \tag{11}$$

$$\boldsymbol{u} = \boldsymbol{u}^h + \boldsymbol{u}', \tag{12}$$

$$p = p^h + p'. \tag{13}$$

Without entering into the details of the derivation, we follow the approach proposed in [2], for which we decompose Eq. (8) into coarse and fine scale equations, we integrate by parts the fine scale terms appearing into the coarse scale equations, and finally we model the fine scale velocity and pressure variables as ([2]):

$$\boldsymbol{u}' \simeq -\tau_M(\boldsymbol{u}^h)\boldsymbol{r}_M(\boldsymbol{u}^h, p^h), \tag{14}$$

$$p' \simeq -\tau_C(\boldsymbol{u}^h) r_C(\boldsymbol{u}^h), \tag{15}$$

where $\boldsymbol{r}_M(\boldsymbol{u}^h, p^h)$ and $r_C(\boldsymbol{u}^h)$ indicate the strong residuals of the momentum and continuity equations:

$$\boldsymbol{r}_M(\boldsymbol{u}^h, p^h) = \rho\frac{\partial \boldsymbol{u}^h}{\partial t} + \rho\boldsymbol{u}^h \cdot \nabla \boldsymbol{u}^h + \nabla p^h - \mu\Delta \boldsymbol{u}^h - \boldsymbol{f}, \tag{16}$$

$$\boldsymbol{r}_C(\boldsymbol{u}^h) = \nabla \cdot \boldsymbol{u}^h, \tag{17}$$

respectively. Moreover, τ_M and τ_C are the stabilization parameters, which we choose similarly to [2] as:

$$\tau_M = \tau_M(\boldsymbol{u}^h) = \left(\frac{\sigma^2\rho^2}{\Delta t^2} + \frac{\rho^2}{h_K^2}|\boldsymbol{u}^h|^2 + \frac{\mu^2}{h_K^4}C_r\right)^{-1/2}, \tag{18}$$

$$\tau_C = \tau_C(\boldsymbol{u}^h) = \frac{h_K^2}{\tau_M(\boldsymbol{u}^h)}, \tag{19}$$

where $C_r = 60 \cdot 2^{r-2}$ is a constant obtained by an inverse inequality relation, σ is a constant equal to the order of the time discretization chosen and Δt is the time step that will be chosen for the time discretization. We remark that the expression chosen for the stabilization parameter τ_M does not account for the projection of the advective and diffusive terms onto the characteristic sizes of the mesh element along the Cartesian components as e.g. in [2,28,48,51], but rather it is "averaged" for simplicity over the mesh element.

Finally, the semi-discrete variational multiscale formulation (VMS-LES) of the Navier–Stokes equations written in terms of the weak residual reads, for all $t \in (0, T)$:

$$\text{find } \boldsymbol{U}^h = \boldsymbol{U}^h(t) = \{\boldsymbol{u}^h, p^h\} \in \boldsymbol{\mathcal{V}}_{\mathbf{g}}^h : A(\boldsymbol{W}^h, \boldsymbol{U}^h) - F(\boldsymbol{W}^h) = 0$$

$$\text{for all } \boldsymbol{W}^h = \{\boldsymbol{w}^h, q^h\} \in \boldsymbol{\mathcal{V}}_{\mathbf{0}}^h, \tag{20}$$

with $\boldsymbol{u}^h(0) = \boldsymbol{u}_0$, where:

$$A(\boldsymbol{W}^h, \boldsymbol{U}^h) := A^{NS}(\boldsymbol{W}^h, \boldsymbol{U}^h) + A^{VMS}(\boldsymbol{W}^h, \boldsymbol{U}^h), \tag{21}$$

$$F(\boldsymbol{W}^h) := (\boldsymbol{w}^h, \boldsymbol{f}) + (\boldsymbol{w}^h, \mathbf{h})_{\Gamma_N}, \tag{22}$$

with:

$$A^{NS}(\boldsymbol{W}^h, \boldsymbol{U}^h) := \left(\boldsymbol{w}^h, \rho\frac{\partial \boldsymbol{u}^h}{\partial t}\right) + (\boldsymbol{w}^h, \rho\boldsymbol{u}^h \cdot \nabla \boldsymbol{u}^h) + \left(\nabla \boldsymbol{w}^h, \mu\left(\nabla \boldsymbol{u}^h + (\nabla \boldsymbol{u}^h)^T\right)\right) - (\nabla \cdot \boldsymbol{w}^h, p^h) + (q^h, \nabla \cdot \boldsymbol{u}^h), \tag{23}$$

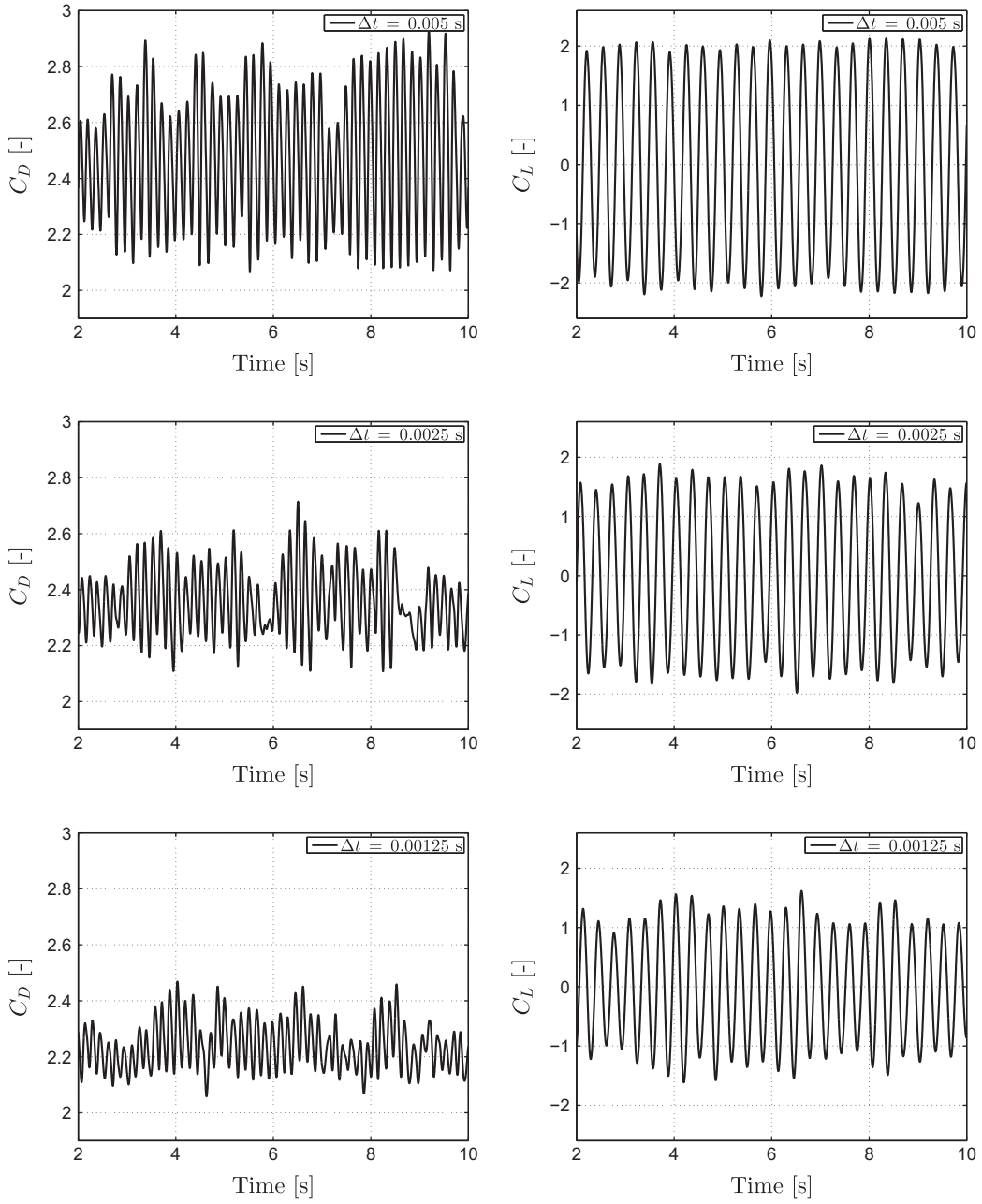


Fig. 4. Results obtained by $\mathbb{P}1$–$\mathbb{P}1$ Finite Element and the BDF1 scheme; C_D (left) and C_L (right) vs. time t [s] for $\Delta t = 0.005$ s (top), 0.0025 s (mid), and 0.00125 s (bottom).

$$\begin{aligned} A^{VMS}(\boldsymbol{W}^h, \boldsymbol{U}^h) := & \left(\rho \boldsymbol{u}^h \cdot \nabla \boldsymbol{w}^h + \nabla q^h, \tau_M(\boldsymbol{u}^h)\boldsymbol{r}_M(\boldsymbol{u}^h, p^h)\right) \\ & - \left(\nabla \cdot \boldsymbol{w}^h, \tau_C(\boldsymbol{u}^h)\boldsymbol{r}_C(\boldsymbol{u}^h)\right) \\ & + \left(\rho \boldsymbol{u}^h \cdot (\nabla \boldsymbol{u}^h)^T, \tau_M(\boldsymbol{u}^h)\boldsymbol{r}_M(\boldsymbol{u}^h, p^h)\right) \\ & - \left(\nabla \boldsymbol{w}^h, \tau_M(\boldsymbol{u}^h)\boldsymbol{r}_M(\boldsymbol{u}^h, p^h) \otimes \tau_M(\boldsymbol{u}^h)\boldsymbol{r}_M(\boldsymbol{u}^h, p^h)\right). \end{aligned} \tag{24}$$

We remark that Eq. (23) defines the standard terms of the Navier–Stokes equations in weak formulation, while Eq. (24) contains the terms obtained from the application of the VMS-LES method. In this respect, the first row of Eq. (24) represents the classical Streamline-Upwind Petrov–Galerkin (SUPG) stabilization terms, while the second and third rows contain terms peculiar of the VMS stabilization; specifically, the second row indicates an additional stabilization term complementing the SUPG ones and finally the third row models the Reynolds cross-stress, which realizes the LES modeling of turbulence [2].

3. Time discretization: semi-implicit BDF schemes

We consider the time discretization of problem (20) by means of the Backward Differentiation Formulas (BDF) [7,16,41], a family of linear multistep methods for which the first order derivative is replaced by a one-sided, high order incremental ratio; such BDF schemes are often used in computational fluid dynamics, as for example in [19].

3.1. Fully implicit BDF schemes

Let us partition the time interval $[0, T]$ into N_t subintervals of equal size $\Delta t = \frac{T}{N_t}$ for which the discrete time instances are $t_n = n\Delta t$ for $n = 0, \ldots, N_t$. Furthermore, let us denote with $\boldsymbol{u}_n^h$ and p_n^h the approximations of the velocity $\boldsymbol{u}^h$ and pressure p^h fields at the time t_n. According to the order σ of the BDF scheme, the

approximation of the time derivative of the velocity appearing in the Navier–Stokes equations reads:

$$\frac{\partial \boldsymbol{u}^h}{\partial t} \approx \frac{\alpha_\sigma \boldsymbol{u}^h_{n+1} - \boldsymbol{u}^h_{n,\mathrm{BDF}\sigma}}{\Delta t}, \tag{25}$$

where for BDF schemes of orders $\sigma = 1, 2, 3$ we have:

$$\boldsymbol{u}^h_{n,\mathrm{BDF}\sigma} = \begin{cases} \boldsymbol{u}^h_n & \text{if } n \geqslant 0, \quad \text{for} \sigma = 1 \quad (\mathrm{BDF1}), \\ 2\boldsymbol{u}^h_n - \frac{1}{2}\boldsymbol{u}^h_{n-1} & \text{if } n \geqslant 1, \quad \text{for} \sigma = 2 \quad (\mathrm{BDF2}), \\ 3\boldsymbol{u}^h_n - \frac{3}{2}\boldsymbol{u}^h_{n-1} + \frac{1}{3}\boldsymbol{u}^h_{n-2} & \text{if } n \geqslant 2, \quad \text{for} \sigma = 3 \quad (\mathrm{BDF3}), \end{cases} \tag{26}$$

and

$$\alpha_\sigma = \begin{cases} 1, & \text{for } \sigma = 1 \quad (\mathrm{BDF1}), \\ \frac{3}{2}, & \text{for } \sigma = 2 \quad (\mathrm{BDF2}), \\ \frac{11}{6}, & \text{for } \sigma = 3 \quad (\mathrm{BDF3}). \end{cases} \tag{27}$$

According to the approximation of the time derivative, Eq. (25) is replaced in Eq. (20), while the other time dependent terms are evaluated at the general discrete time instance t_{n+1}. In this way, after the spatial and time discretizations, the fully discrete formulation of problem (20) reads at time t_{n+1} and for a given BDF scheme of order σ:

$$\begin{aligned} &\text{find } \boldsymbol{u}^h_{n+1} \in \mathcal{V}^h_{\mathbf{g}} \text{ and } p^h_{n+1} \in \mathcal{Q}^h \\ &: \left(\boldsymbol{w}^h, \rho \frac{\alpha_\sigma \boldsymbol{u}^h_{n+1} - \boldsymbol{u}^h_{n,\mathrm{BDF}\sigma}}{\Delta t}\right) + \left(\boldsymbol{w}^h, \rho \boldsymbol{u}^h_{n+1} \cdot \nabla \boldsymbol{u}^h_{n+1}\right) \\ &+ \left(\nabla \boldsymbol{w}^h, \mu\left(\nabla \boldsymbol{u}^h_{n+1} + (\nabla \boldsymbol{u}^h_{n+1})^T\right)\right) - \left(\nabla \cdot \boldsymbol{w}^h, p^h_{n+1}\right) \\ &+ \left(q^h, \nabla \cdot \boldsymbol{u}^h_{n+1}\right) \\ &+ \left(\rho \boldsymbol{u}^h_{n+1} \cdot \nabla \boldsymbol{w}^h + \nabla q^h, \tau_M(\boldsymbol{u}^h_{n+1}) \check{\boldsymbol{r}}_M(\boldsymbol{u}^h_{n+1}, p^h_{n+1})\right) \\ &- \left(\nabla \cdot \boldsymbol{w}^h, \tau_C(\boldsymbol{u}^h_{n+1}) \boldsymbol{r}_C(\boldsymbol{u}^h_{n+1})\right) \\ &+ \left(\rho \boldsymbol{u}^h_{n+1} \cdot (\nabla \boldsymbol{w}^h)^T, \tau_M(\boldsymbol{u}^h_{n+1}) \check{\boldsymbol{r}}_M(\boldsymbol{u}^h_{n+1}, p^h_{n+1})\right) \\ &- \left(\nabla \boldsymbol{w}^h, \tau_M(\boldsymbol{u}^h_{n+1}) \check{\boldsymbol{r}}_M(\boldsymbol{u}^h_{n+1}, p^h_{n+1}) \otimes \tau_M(\boldsymbol{u}^h_{n+1}) \check{\boldsymbol{r}}_M(\boldsymbol{u}^h_{n+1}, p^h_{n+1})\right) \\ &= \left(\boldsymbol{w}^h, \boldsymbol{f}_{n+1}\right) + \left(\boldsymbol{w}^h, \mathbf{h}_{n+1}\right)_{\Gamma_N}, \text{for all } \boldsymbol{w}^h \in \mathcal{V}^h_{\mathbf{0}} \text{ and } q^h \\ &\in \mathcal{Q}^h, \text{ for all } n \geqslant \sigma - 1, \end{aligned} \tag{28}$$

given $\boldsymbol{u}^h_n, \ldots, \boldsymbol{u}^h_{n+1-\sigma}$, with $\boldsymbol{f}_{n+1} = \boldsymbol{f}(t_{n+1})$, $\mathbf{h}_{n+1} = \mathbf{h}(t_{n+1})$ $\check{\boldsymbol{r}}_M(\boldsymbol{u}^h_{n+1}, p^h_{n+1}) := \rho\left(\frac{\alpha_\sigma \boldsymbol{u}^h_{n+1} - \boldsymbol{u}^h_{n,\mathrm{BDF}\sigma}}{\Delta t}\right) + \rho \boldsymbol{u}^h_{n+1} \cdot \nabla \boldsymbol{u}^h_{n+1} + \nabla p^h_{n+1} - \mu \Delta \boldsymbol{u}^h_{n+1} - \boldsymbol{f}_{n+1}$.

We notice that the BDF1 scheme coincides with the Backward Euler method. In this work, we consider a fixed time step Δt; nevertheless, it is possible to use a time varying or adaptive time step compatible with the BDF schemes, as done e.g. in [9,53].

At each discrete time t_n, the BDF scheme yields a nonlinear problem to be solved, since (28) is nonlinear in the variables $\boldsymbol{u}^h_{n+1}$ and p^h_{n+1}. An approximation of this nonlinear problem can be obtained for example with the Newton method [41]. This requires, at each Newton iterate, the assembly of the Jacobian matrix and the solution of a linear system. A fully implicit approach, although obtained for different time discretizations, has been widely used for the study of incompressible fluid flows in turbulent regimes as e.g. in [2,25,37]. However, while a fully implicit approach is generally yielding a stable time discretization scheme, the associated computational costs may be significantly high due to the repeated assembly of the residual vector and Jacobian matrix and the solution of the associated linear system.

3.2. Semi-implicit BDF schemes

In this paper, in order to contain the computational burden associated to the use of a fully implicit BDF approach (28), we consider instead a semi-implicit BDF scheme derived from problem (28), for which the nonlinear terms in $\boldsymbol{u}^h_{n+1}$ and p^h_{n+1} are extrapolated by means of the Newton–Gregory backward polynomials [10,44]. Without entering into the details of the derivation, for which we refer the reader to e.g. [10,44], we consider the following extrapolations of orders $\sigma = 1, 2, 3$ for the velocity and pressure variables at the discrete time t_{n+1}:

$$\boldsymbol{u}^h_{n+1,\sigma} = \begin{cases} \boldsymbol{u}^h_n & \text{if } n \geqslant 0, \quad \text{for } \sigma = 1 \quad (\mathrm{BDF1}), \\ 2\boldsymbol{u}^h_n - \boldsymbol{u}^h_{n-1} & \text{if } n \geqslant 1, \quad \text{for } \sigma = 2 \quad (\mathrm{BDF2}), \\ 3\boldsymbol{u}^h_n - 3\boldsymbol{u}^h_{n-1} + \boldsymbol{u}^h_{n-2} & \text{if } n \geqslant 2, \quad \text{for } \sigma = 3 \quad (\mathrm{BDF3}), \end{cases} \tag{29}$$

and similarly:

$$p^h_{n+1,\sigma} = \begin{cases} p^h_n & \text{if } n \geqslant 0, \quad \text{for } \sigma = 1 \quad (\mathrm{BDF1}), \\ 2p^h_n - p^h_{n-1} & \text{if } n \geqslant 1, \quad \text{for } \sigma = 2 \quad (\mathrm{BDF2}), \\ 3p^h_n - 3p^h_{n-1} + p^h_{n-2} & \text{if } n \geqslant 2, \quad \text{for } \sigma = 3 \quad (\mathrm{BDF3}). \end{cases} \tag{30}$$

Starting from the fully implicit formulation (28), we use the above extrapolations by Newton–Gregory backward polynomials. In this way, for a given BDF scheme of order σ, the fully discrete linearized semi-implicit formulation of problem (28) reads:

$$\begin{aligned} &\text{find } \boldsymbol{u}^h_{n+1} \in \mathcal{V}^h_{\mathbf{g}} \text{ and} p^h_{n+1} \in \mathcal{Q}^h \\ &: \left(\boldsymbol{w}^h, \rho \frac{\alpha_\sigma \boldsymbol{u}^h_{n+1} - \boldsymbol{u}^h_{n,\mathrm{BDF}\sigma}}{\Delta t}\right) + \left(\boldsymbol{w}^h, \rho \boldsymbol{u}^h_{n+1,\sigma} \cdot \nabla \boldsymbol{u}^h_{n+1}\right) \\ &+ \left(\nabla \boldsymbol{w}^h, \mu\left(\nabla \boldsymbol{u}^h_{n+1} + (\nabla \boldsymbol{u}^h_{n+1})^T\right)\right) - \left(\nabla \cdot \boldsymbol{w}^h, p^h_{n+1}\right) \\ &+ \left(q^h, \nabla \cdot \boldsymbol{u}^h_{n+1}\right) \\ &+ \left(\rho \boldsymbol{u}^h_{n+1,\sigma} \cdot \nabla \boldsymbol{w}^h + \nabla q^h, \tau_M^{n+1,\sigma} \boldsymbol{r}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1}, p^h_{n+1})\right) \\ &- \left(\nabla \cdot \boldsymbol{w}^h, \tau_C^{n+1,\sigma} \boldsymbol{r}_C(\boldsymbol{u}^h_{n+1})\right) \\ &+ \left(\rho \boldsymbol{u}^h_{n+1,\sigma} \cdot (\nabla \boldsymbol{w}^h)^T, \tau_M^{n+1,\sigma} \boldsymbol{r}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1}, p^h_{n+1})\right) \\ &- \left(\nabla \boldsymbol{w}^h, \tau_M^{n+1,\sigma} \widehat{\boldsymbol{r}}_M^{n+1,\sigma} \otimes \tau_M^{n+1,\sigma} \widetilde{\boldsymbol{r}}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1}, p^h_{n+1})\right) \\ &- \left(\nabla \boldsymbol{w}^h, \tau_M^{n+1,\sigma} \widehat{\boldsymbol{r}}_M^{n+1,\sigma} \otimes \tau_M^{n+1,\sigma} \rho \alpha_\sigma \frac{\boldsymbol{u}^h_{n+1}}{\Delta t}\right) \\ &+ \left(\nabla \boldsymbol{w}^h, \tau_M^{n+1,\sigma} \boldsymbol{r}_M^{n+1,\sigma}(\boldsymbol{u}^h_{n+1}, p^h_{n+1}) \otimes \tau_M^{n+1,\sigma} \rho \frac{\boldsymbol{u}^h_{n,BDF\sigma}}{\Delta t}\right) \\ &= \left(\boldsymbol{w}^h, \boldsymbol{f}_{n+1}\right) + \left(\boldsymbol{w}^h, \mathbf{h}_{n+1}\right)_{\Gamma_N}, \quad \text{for all } \boldsymbol{w}^h \\ &\in \mathcal{V}^h_{\mathbf{0}} \text{ and } q^h \in \mathcal{Q}^h, \text{ for all } n \geqslant \sigma - 1, \end{aligned} \tag{31}$$

given $\boldsymbol{u}^h_n, \ldots, \boldsymbol{u}^h_{n+1-\sigma}$, where the stabilization parameters are defined as follows:

$$\tau_M^{n+1,\sigma} := \left(\frac{\sigma^2 \rho^2}{\Delta t^2} + \frac{\rho^2}{h_K^2}\left|\boldsymbol{u}^h_{n+1,\sigma}\right|^2 + \frac{\mu^2}{h_K^4} C_r\right)^{-1/2}, \tag{32}$$

$$\tau_C^{n+1,\sigma} = \frac{h_K^2}{\tau_M^{n+1,\sigma}}, \tag{33}$$

and the residuals read:

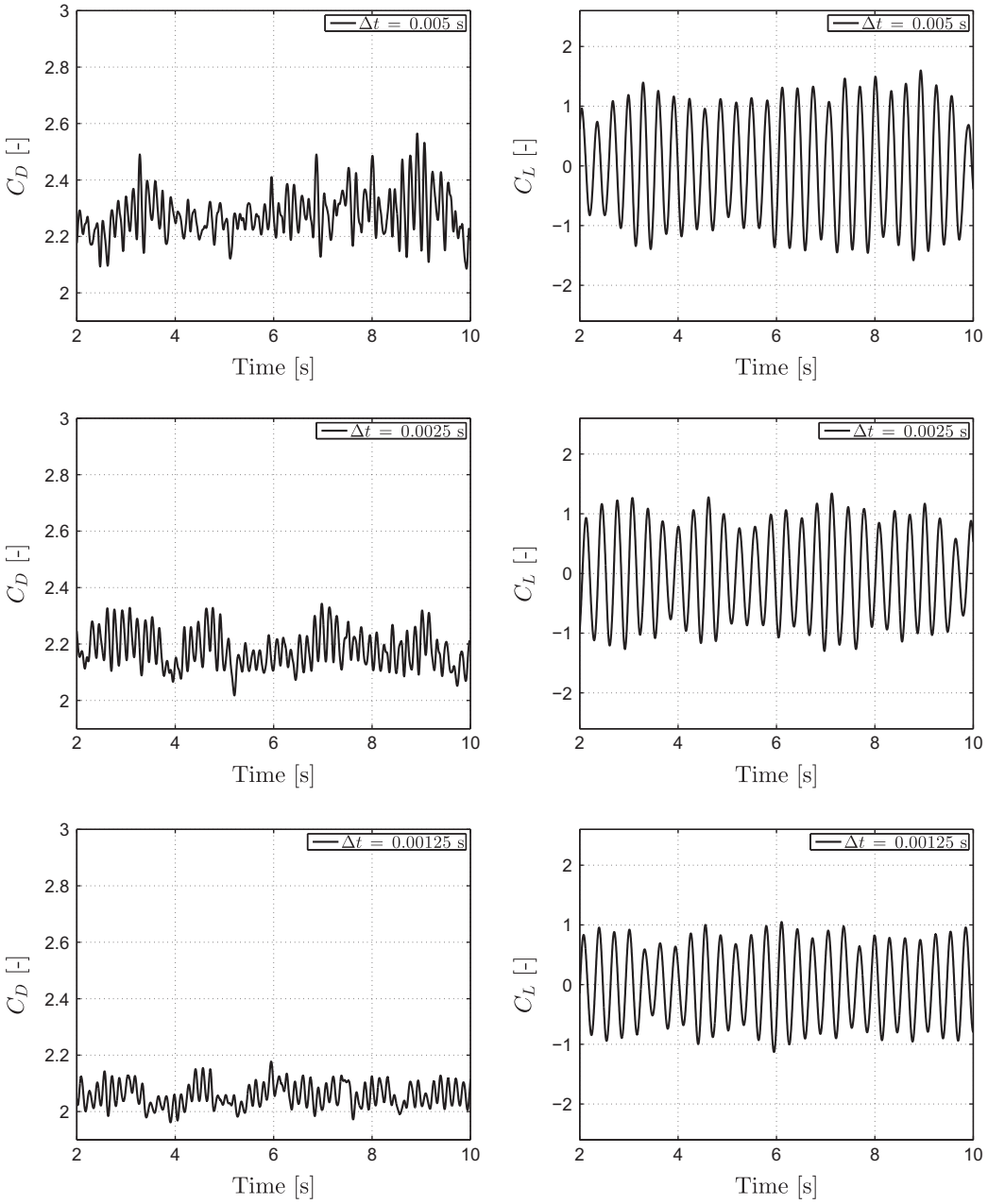


Fig. 5. Results obtained by $\mathbb{P}1$–$\mathbb{P}1$ Finite Element and the BDF2 scheme; C_D (left) and C_L (right) vs. time t [s] for $\Delta t = 0.005$ s (top), 0.0025 s (mid), and 0.00125 s (bottom).

$$
\begin{aligned}
&\boldsymbol{r}_M^{n+1,\sigma}(\boldsymbol{u}_{n+1}^h, p_{n+1}^h) := \rho\left(\frac{\alpha_\sigma \boldsymbol{u}_{n+1}^h - \boldsymbol{u}_{n,\mathrm{BDF}\sigma}^h}{\Delta t}\right)\\
&\quad + \rho \boldsymbol{u}_{n+1,\sigma}^h \cdot \nabla \boldsymbol{u}_{n+1}^h + \nabla p_{n+1}^h - \mu \Delta \boldsymbol{u}_{n+1}^h - \boldsymbol{f}_{n+1},\\
&\widehat{\boldsymbol{r}}_M^{n+1,\sigma} := \boldsymbol{r}_M^{n+1,\sigma}(\boldsymbol{u}_{n+1,\sigma}^h, p_{n+1,\sigma}^h),\\
&\widetilde{\boldsymbol{r}}_M^{n+1,\sigma}(\boldsymbol{u}_{n+1}^h, p_{n+1}^h) := \rho \boldsymbol{u}_{n+1,\sigma}^h \cdot \nabla \boldsymbol{u}_{n+1}^h + \nabla p_{n+1}^h - \mu \Delta \boldsymbol{u}_{n+1}^h - \boldsymbol{f}_{n+1}.
\end{aligned}
\tag{34}
$$

Thanks to the time discretization proposed, the fully discrete semi-implicit formulation (31) yields a linear problem in the variables $\boldsymbol{u}_{n+1}^h$ and p_{n+1}^h to be solved only once at each time t_n. We remark that the Newton–Gregory extrapolation of the pressure variable in Eq. (30) is required by the terms of the formulation which carry the LES modeling. In this respect, we notice that a linearization of Eq. (28) by means of first order Frechét differentiation would have led to a semi-implicit formulation with a larger number of terms than in Eq. (31), thus resulting in a larger assembly cost than in the proposed semi-implicit BDF formulation.

Finally, we remark that semi-implicit BDF methods with Newton–Gregory extrapolation yield stable time discretization schemes if the velocity field is divergence free a.e. in the computational domain. However, this requirement is not guaranteed for the spatially approximated velocity field $\boldsymbol{u}_n^h$ obtained by solving the Navier–Stokes equations in the VMS-LES formulation, which in principle would require restrictions on the choice of the time step. Nevertheless, our numerical experience indicates that the use of relatively "large" time steps still yields stable solutions for the proposed semi-implicit VMS-LES method with BDF time discretization.

4. Linear parallel solver: GMRES with multigrid preconditioner

We implement the semi-implicit VMS-LES method with BDF time discretization in the open-source Finite Element library *LifeV* [55] for the High Performance Computing of problems described by PDEs. In our numerical tests, the linear system arising from Eq. (31) is solved in a parallel setting using the GMRES method preconditioned by an algebraic three levels multigrid preconditioner based on the ML package of *Trilinos* [17]; see e.g. [42,52]. For the application of the multigrid (ML) right preconditioner to the matrix we perform three sweeps of the Gauss–Seidel algorithm for pre- and post-smoothing, while the solution on the coarsest level is based on a *LU* factorization [41]. In our computations, we consider the relative residual as stopping criterion for the GMRES method with tolerance equal to 10^{-6}.

5. Numerical results

In this section we validate the semi-implicit BDF VMS-LES scheme towards the benchmark problem of the vortex shedding induced by the fluid flow past a squared cylinder at high Reynolds number [25]. The simulations have been carried out for different orders of both the time and spatial discretizations and for different time steps with the aim of studying their influence on the accuracy of the numerical solution. We discuss the numerical performance of the solver and analyze the influence of the stabilization parameters on the numerical results.

5.1. The benchmark problem: flow past a squared cylinder

As a validation test, we numerically simulate the flow past a squared cylinder at Reynolds number $Re = 22,000$. This problem

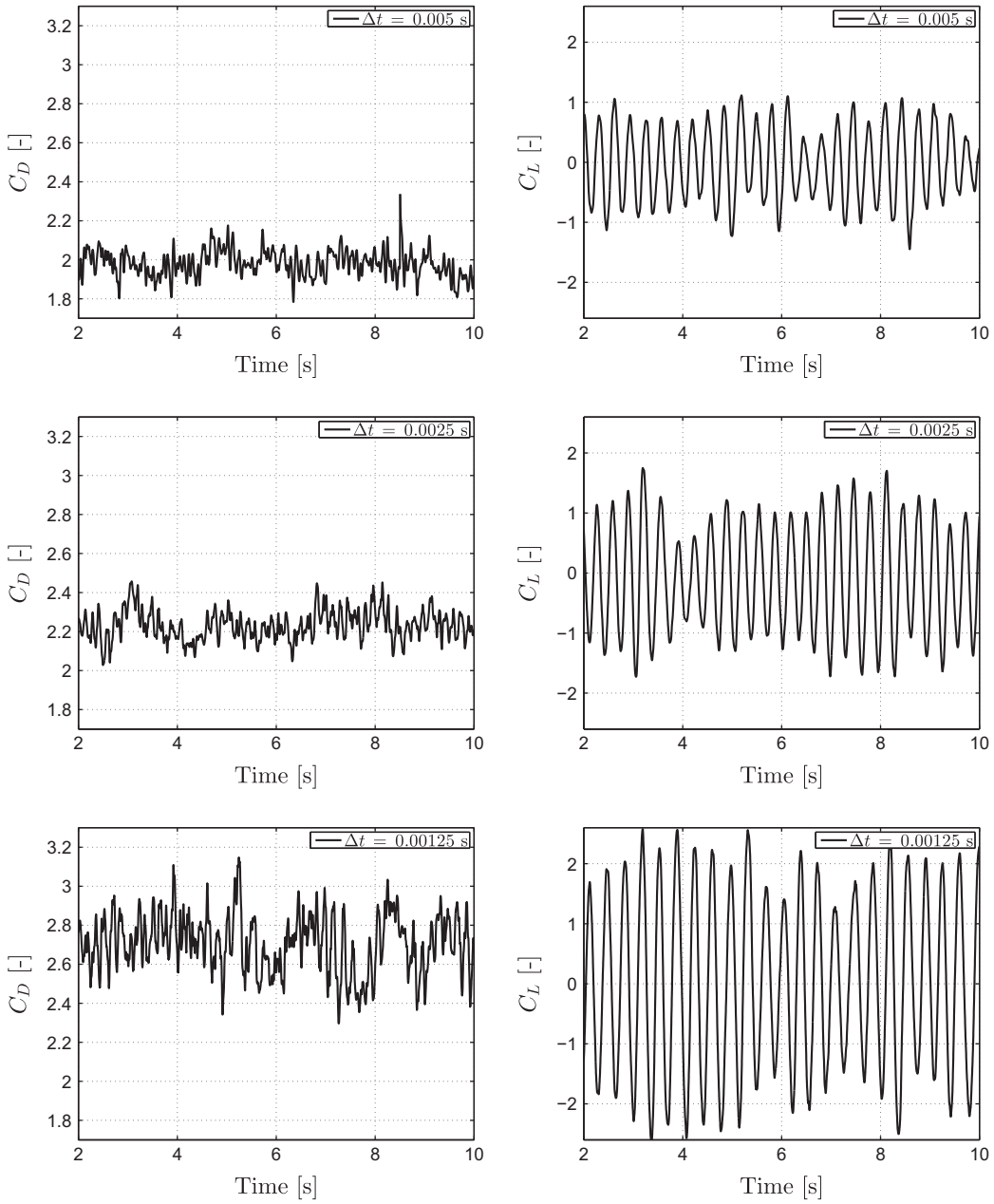


Fig. 6. Results obtained by $\mathbb{P}2$–$\mathbb{P}2$ Finite Element and the BDF2 scheme; C_D (left) and C_L (right) vs. time t [s] for $\Delta t = 0.005$ s (top), 0.0025 s (mid), and 0.00125 s (bottom).

Table 2
Results obtained by the semi-implicit BDF VMS-LES method by considering different time step s Δt and orders for the spatial and time discretizations.

FE discretization	Time step	BDF order	$\overline{C}_D$	rms(C_D)	rms(C_L)	Strouhal
$\mathbb{P}1$–$\mathbb{P}1$	$\Delta t = 0.01$ s	1	2.51	0.250	1.59	0.127
	$\Delta t = 0.005$ s	1	2.49	0.226	1.49	0.133
	$\Delta t = 0.0025$ s	1	2.35	0.112	1.18	0.138
	$\Delta t = 0.00125$ s	1	2.24	0.0785	0.895	0.142
	$\Delta t = 0.01$ s	2	2.33	0.204	1.11	0.142
	$\Delta t = 0.005$ s	2	2.27	0.0896	0.897	0.144
	$\Delta t = 0.0025$ s	2	2.16	0.0622	0.658	0.146
	$\Delta t = 0.00125$ s	2	2.05	0.0420	0.576	0.146
$\mathbb{P}2$–$\mathbb{P}2$	$\Delta t = 0.005$ s	2	1.98	0.102	0.580	0.142
	$\Delta t = 0.0025$ s	2	2.24	0.121	0.986	0.141
	$\Delta t = 0.00125$ s	2	2.71	0.153	1.49	0.129

Table 3
Results obtained by different numerical simulations based on LES models (results between brackets indicate a range of values).

LES method	$\overline{C}_D$	rms(C_D)	rms(C_L)	Strouhal
Finite volumes VMS (Koobus et al. [25])	2.10	0.18	1.08	0.136
Smagorinsky (Rodi et al. [45])	[1.66–2.77]	[0.1–0.27]	[0.38–1.79]	[0.07–0.15]
Dynamic LES (Sohankar et al. [50])	[2.00–2.32]	[0.16–0.20]	[1.23–1.54]	[0.127–0.135]

has been already investigated both experimentally [5,36] and numerically by using Smagorinsky and dynamic LES models [45,50], as well as with a VMS formulation for the compressible Euler equations approximated by the Finite Volumes method [25] (see also [26]).

The geometrical setting considered for this benchmark problem is schematically illustrated in Fig. 1, for which $D = 1$ m is the side length of the squared cylinder, $L_{in} = 4.5$ D and $L_{out} = 15.5$ D are the distances between the cylinder and the inflow and outflow surface boundaries, respectively, $L_s = 4$ D is the width of the domain and $H = 6.5$ D is the distance between the cylinder and the bottom and top walls.

Regarding the boundary conditions, at the surfaces on the top and bottom of the domain we set a null normal component of the velocity vector $\boldsymbol{u} \cdot \hat{\mathbf{n}} = 0$; similarly, at the lateral boundaries we set the normal velocity component $\boldsymbol{u} \cdot \hat{\mathbf{n}} = 0$. At the inflow of the domain Γ_{in}, we prescribe a velocity profile $\boldsymbol{u}_{in}(t)$ that is uniform along the inflow section and it is dependent only on the time independent variable as:

$$\boldsymbol{u}_{in}(t) = \begin{cases} \frac{V_\infty}{2}\left(1 - \cos\left(\pi \frac{t}{T_r}\right)\right) & \text{if } 0 \leqslant t < T_r, \\ V_\infty & \text{if } t \geqslant T_r, \end{cases}$$

where, in the numerical tests, we have consider $T_r = 0.3$ s and the reference inflow velocity $V_\infty = 22$ m/s At the outflow boundary Γ_{out}, we consider the following natural boundary condition [3]:

$$-p\hat{\mathbf{n}} + 2\mu(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T) \cdot \hat{\mathbf{n}} - \rho(\{\boldsymbol{u} \cdot \hat{\mathbf{n}}\}_-)\boldsymbol{u} = 0 \quad \text{on } \Gamma_{out}, \tag{35}$$

where $\hat{\mathbf{n}}$ is the outward directed unit vector normal to Γ_{out} and $\{\boldsymbol{u} \cdot \hat{\mathbf{n}}\}_-$ denotes the negative part of $\boldsymbol{u} \cdot \hat{\mathbf{n}}$:

$$\{\boldsymbol{u} \cdot \hat{\mathbf{n}}\}_- = \begin{cases} \boldsymbol{u} \cdot \hat{\mathbf{n}} & \text{if } \boldsymbol{u} \cdot \hat{\mathbf{n}} < 0, \\ 0 & \text{if } \boldsymbol{u} \cdot \hat{\mathbf{n}} \geqslant 0. \end{cases}$$

The above boundary condition is introduced to weakly penalize the reverse flow induced by the vortexes at Γ_{out}, which may render unstable the discrete formulation of the problem. Indeed, we observe that only if $\boldsymbol{u} \cdot \hat{\mathbf{n}} < 0$ on Γ_{out}, the last term in the left hand side of Eq. (35) is active; if $\boldsymbol{u} \cdot \hat{\mathbf{n}} \geqslant 0$ Γ_{out}, the outflow boundary condition considered coincides with the well known stress-free condition. On the squared cylinder surface we impose a no slip boundary condition ($\boldsymbol{u} = \mathbf{0}$). We consider the fluid to have density $\rho = 1000$ kg/m^3 and a dynamic viscosity $\mu = 1$ Pa s. In this way the Reynolds number is $Re = \frac{\rho V_\infty D}{\mu} = 22,000$.

We discretize the computational domain Ω by a mesh comprised of 330,764 vertices and 1,853,500 tetrahedral elements. In Fig. 2, two images of the computational mesh $\mathcal{T}_h$ are presented: on the left we show a vertical cut plane of the mesh and on the right an horizontal cut through the squared cylinder. Along the side length of the cylinder we considered about 20 mesh elements, corresponding to a length $h_{bl} = 0.05$ m. We remark that the mesh size is not uniform in Ω and the ratio between the largest and the smallest local mesh sizes is about 10–15.

We solve the benchmark problem using the semi-implicit VMS-LES method based on the BDF formulas. We analyze the performances of the method with respect to the use of first ($\sigma = 1$) and second ($\sigma = 2$) order BDF schemes for different time steps ($\Delta t = 0.005s, \Delta t = 0.0025$ s, and $\Delta t = 0.00125$ s) and different degrees of polynomials used for the Finite Element approximation, specifically $\mathbb{P}1$–$\mathbb{P}1$ for $r = 1$ and $\mathbb{P}2$–$\mathbb{P}2$ for $r = 2$. The time interval considered for the simulations is such that $t \in (0, 14)$ s. In Table 1 we report the number of degrees of freedom associated to the discretized problem for the numerical simulations performed.

In Fig. 3 we show a post-processing of the solution obtained by using $\mathbb{P}2$–$\mathbb{P}2$ Finite Element and BDF2 scheme at time $t = 10$ s for $\Delta t = 0.0025$ s, i.e. when the turbulent flow is fully developed. Fig. 3(a) shows the coherent vortex structures characterizing the wake (we identified the structures by means of the λ_2 criterion [23]); in Fig. 3(b), we highlight the considerable three-dimensional features of the turbulent flow occurring in the wake region adjacent to the cylinder by means of a representation of the streamlines colored by the vorticity field. Fig. 3(c) illustrates the pressure field at a plane located at $z = 0$, as well as some significant isosurfaces; we showed only negative values of the pressure field in order to highlight the low pressure zones characterizing the center of the vortexes in the wake region. In Fig. 3(d) we illustrate the vorticity field, both on the squared cylinder and on a cut plane at $z = 0$, on which we also show, by means of a surface LIC, the recirculation of the flow detaching from the cylinder.

In order to compare the results with those available in literature obtained by other LES methods, we compute the drag and lift

coefficients on the cylinder. Let us introduce $\hat{\mathbf{v}}_\infty = \frac{\mathbf{v}_\infty}{\|\mathbf{v}_\infty\|}$, that is a unit vector directed as the incoming flow, and $\hat{\mathbf{n}}_\infty$, a unit vector orthogonal to the direction $\hat{\mathbf{v}}_\infty$ of the incoming flow. The aerodynamic drag and lift coefficients for the cylinder are defined as:

$$C_D(\boldsymbol{u},p) = -\frac{1}{q_\infty |\Gamma_{BODY}|}\oint_{\Gamma_{BODY}} (\boldsymbol{\sigma}(\boldsymbol{u},p)\hat{\mathbf{n}})\cdot\hat{\mathbf{v}}_\infty d\Gamma, \tag{36}$$

$$C_L(\boldsymbol{u},p) = \frac{1}{q_\infty |\Gamma_{BODY}|}\oint_{\Gamma_{BODY}} (\boldsymbol{\sigma}(\boldsymbol{u},p)\hat{\mathbf{n}})\cdot\hat{\mathbf{n}}_\infty d\Gamma, \tag{37}$$

where $q_\infty = \frac{1}{2}\rho V_\infty^2$ is the dynamic pressure, and $|\Gamma_{BODY}|$ is the surface area of the cylinder; in Eq. (36) the minus sign takes into account that, by convention, the force is positive if acting on the fluid. In practice, we compute the aerodynamic drag and lift coefficients by means of the weak residual form ([6,14,27]) which ensures higher accuracy than the direct use of Eqs. (36) and (37); with this aim, we use the test functions $\mathbf{d}_\infty \in \{[H^1(\Omega)]^d : \mathbf{d}_\infty|_{\Gamma_{BODY}} = \hat{\mathbf{v}}_\infty, \mathbf{d}_\infty|_{\partial\Omega\setminus\Gamma_{BODY}} = \mathbf{0}\}$ and $\mathbf{l}_\infty \in \{[H^1(\Omega)]^d : \mathbf{l}_\infty|_{\Gamma_{BODY}} = \hat{\mathbf{n}}_\infty, \mathbf{l}_\infty|_{\partial\Omega\setminus\Gamma_{BODY}} = \mathbf{0}\}$ in Eq. (28). In Fig. 4 we illustrate, for the different time steps considered, the evolution of the drag (left-hand side) and lift (right-hand side) coefficients when using a first order scheme for both the spatial and the time discretizations. In the same way, Figs. 5 and 6 show the evolution of the coefficients using a second order BDF scheme in time and a first and second order approximations in space, respectively. By a comparison of the numerical results obtained using $\mathbb{P}1$–$\mathbb{P}1$ Finite Element (Figs. 4 and 5) we highlight the effects of the time discretization. In detail, we can observe that the amplitude of the oscillations of the lift and drag coefficients are substantially reduced when using a second order time discretization with respect to BDF1. In the same way, the results of Figs. 5 and 6 (both obtained using the BDF2) allows to analyze the effects of spatial discretization on the solution. More specifically, it is interesting to analyze the interplay between the choice of the time stepΔt and the order of the spatial discretization. When using $\mathbb{P}1$–$\mathbb{P}1$ Finite Element, the smaller Δt reduces both the amplitudes of the oscillations of lift and drag coefficients, as well as the mean drag coefficients $\overline{C}_D$. On the contrary, when a second order spatial discretization is adopted, smaller time steps lead to wider oscillations of the coefficients and to higher mean values of the mean of the drag.

In order to compare the numerical results with those available in literature, we evaluate the mean drag coefficient $\overline{C}_D$, the root mean square of the lift and of the drag coefficients, say rms(C_D) and rms(C_L), and the Strouhal number obtained from a Fourier analysis of the lift coefficient. In Table 2 we report the results predicted by the semi-implicit BDF VMS-LES method, whereas Table 3 contains the values obtained by other LES methods: in particular, we consider the results obtained with the VMS-Finite Volumes method of Koobus et al. [25], LES models of Rodi et al. [45], and Smagorinsky models of Sohankar et al. [50].

The choice of the time step Δt for the simulations has an important effect on the aerodynamic coefficients independently from the order of the spatial and time discretizations. Indeed, from Table 2 we observe that the estimation of the mean drag coefficient $\overline{C}_D$ obtained with $\Delta t = 0.005$ s and $\Delta t = 0.00125$ s varies of approximately $\pm 5\%$ with respect to the value obtained with $\Delta t = 0.0025$ s. We notice that the use of $\mathbb{P}1$–$\mathbb{P}1$ Finite Element yields a mean drag coefficient $\overline{C}_D$ that decreases as the time step gets smaller; on the contrary, by using $\mathbb{P}2$–$\mathbb{P}2$ Finite Element to approximate the velocity and pressure fields, the value of $\overline{C}_D$ increases as Δt decreases. The same consideration follows from the analysis of the root mean square values of both the lift and drag coefficients. The results obtained using a first order spatial-time discretizations ($\mathbb{P}1$–$\mathbb{P}1$ and BDF1) are in line with those available in literature: although the largest time step under consideration leads already to an accurate prediction of the Strouhal number, the smallest one, i.e. $\Delta t = 0.00125$ s, yields good estimation of all the coefficients. We notice that the use of a first order discretization in space and second in time ($\mathbb{P}1$–$\mathbb{P}1$ and BDF2) leads to an underestimation of the root mean square of the lift and drag coefficients with respect to the results available in literature for LES models, although the prediction of $\overline{C}_D$ is sufficiently accurate. Finally, by employing a second order spatial-time discretization ($\mathbb{P}2$–$\mathbb{P}2$ and BDF2) we notice that the rms(C_L) and rms(C_D) are in line with those obtained with other LES methods, although the smallest time step yields an overestimation of the mean drag coefficient $\overline{C}_D$.

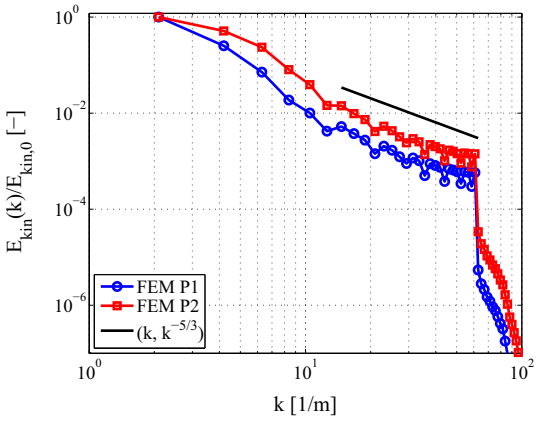


Fig. 7. Relative energy spectra (energy vs. wave number) obtained using a first and second order spatial discretization for BDF2 and $\Delta t = 0.0025$ s; logarithmic scales on both axes. The analysis is performed at time $t = 10$ s, when turbulence is fully developed, and has been carried out by using the software *AnaFlame* [54].

5.2. Energy spectra

It is well known [40] that turbulence features a wide range of spatial and time scales. Fluctuation of the energy is produced at the level of large eddies (with low wave numbers). Vortex stretching mechanisms then generate smaller and smaller eddies and energy "drifts" down the spectrum to higher wave numbers. The slope of the energy spectrum vs. the wave number in this range remains constant, as Kolmogorov showed that this slope is $-5/3$, based on dimensional arguments. We denote by $E_{kin}(k)$ the energy spectrum, where k is the wave number (inversely proportional to the wavelength). In order to compute the energy spectrum, the numerical solution (velocity field) obtained on the squared cylinder benchmark problem is evaluated on a lattice of 10^6 equally spaced points. The points are located within a cube in the wake, centered in 5 m downstream the cylinder. The side length of the cube is $L = 3$ m and 10^2 points are equally spaced along each direction; the results displayed in Fig. 7 show the energy spectra obtained. The slope of the energy spectra,[1] for both the spatial discretizations considered, is in line with the one of the theoretical curve $(k, k^{-5/3})$.

5.3. Parallel performances of the solver

In Fig. 8 we report a strong scalability study of the solver performed using the $\mathbb{P}2$–$\mathbb{P}2$ and BDF2 discretizations of the benchmark problem for $\Delta t = 0.0025$ s. We recall that in this settings

[1] The sharp drop of E_{kin} beyond $k \simeq 401/m$ is due to the fact that the local size h_K of the computational mesh is larger than the distance of the points used for the sampling of the velocity field, for which the associated energy is negligible.

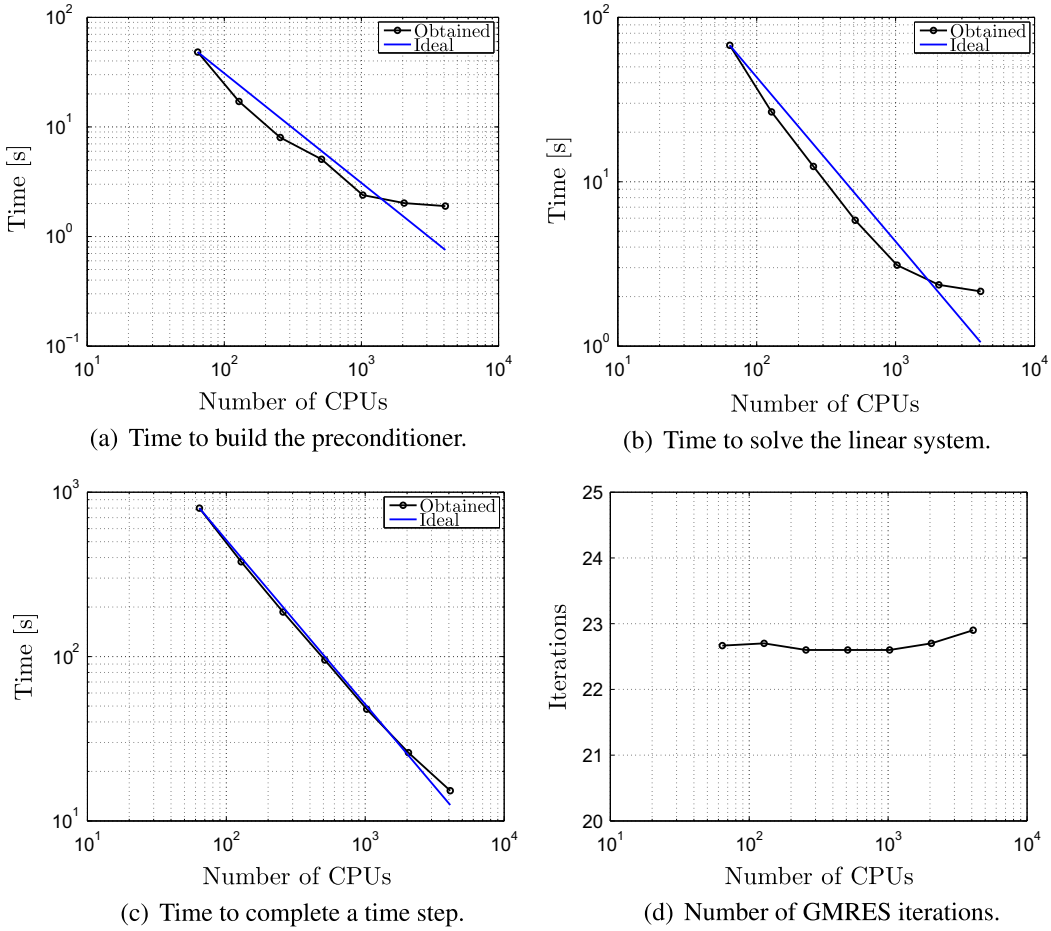
the total number of degrees of freedom is equal to $9,209,040$. In our investigation, we consider a varying number of CPUs ranging from 64 to 4096 and we monitor the time to build the preconditioner (Fig. 8(a)), to solve the linear system by the preconditioned GMRES method (Fig. 8(b)), and to perform a time step, i.e. the sum of the time spent to update the convective and the VMS-LES terms in the assembly phase, to build the preconditioner, and to solve the linear system (Fig. 8(c)). Furthermore, in Fig. 8(d) we show the mean number of GMRES iterations employed to solve the linear system with the prescribed tolerance. All the computations are carried out using `Piz Dora`, a Cray XC40 supercomputer installed at the Swiss National Supercomputing Center (CSCS) whose main technical specifications are reported in Table 4.

The results reported in Fig. 8 show the excellent scalability properties of the solver up to 2048 CPUs: in fact, all the indicators scale almost perfectly and the number of iterations of the GMRES solver remains constant. More specifically, only when using 2048 CPUs we start to observe an initial deterioration in the scalability of the time to build the preconditioner and to solve the linear system. Nevertheless, in terms of time to perform a time step, remarkably only with 4096 CPUs we start to notice a slight deterioration of the performance. This is due to the fact that, within the overall count of the operations required to perform a time step, the assembly phase necessary to update the convective and VMS-LES stabilization terms is the most expensive, taking around the 85% of the total time spent for a time step, although perfectly scalable. This follows from the fact that the VMS-LES formulation (31) requires the assembly of several terms, in particular those associated to the second order Laplace operator; as consequence, this results in an increased computational cost with respect to formulations of the Navier–Stokes equations without the LES modeling terms.

Regarding the GMRES solver, in order to asses the suitability of the tolerance 10^{-6} for the stopping criterion based on the relative residual, we solve the problem also with the tolerance 10^{-4}; with this aim, we use the first order discretization in space and second in time ($\mathbb{P}1$–$\mathbb{P}1$ and BDF2) for $\Delta t = 0.0025$ s. In this case, the computational cost associated to the GMRES solver is greatly reduced with respect to the tolerance 10^{-6}, as the number of iterations is about halved. At the same time, the numerical results are in line with those obtained previously, with a discrepancy of the mean drag coefficient $\overline{C}_D$ of about 1%. Conversely, reducing the tolerance under 10^{-6} would lead to an unnecessary increase of the computational cost associated to the GMRES solver without any noticeable improvement of the numerical results.

Table 4
`Piz Dora` Cray XC40 technical data.

Number of compute nodes	1256
Processor	Dual-socket Intel Xeon processor E5-2690v3, 2.5 GHz
Memory	64 GB per node in 1192 nodes 128 GB per node in 64 nodes
Memory bandwidth	Up to 137 GB/s per node
Network	Dragonfly interconnect



(a) Time to build the preconditioner.

(b) Time to solve the linear system.

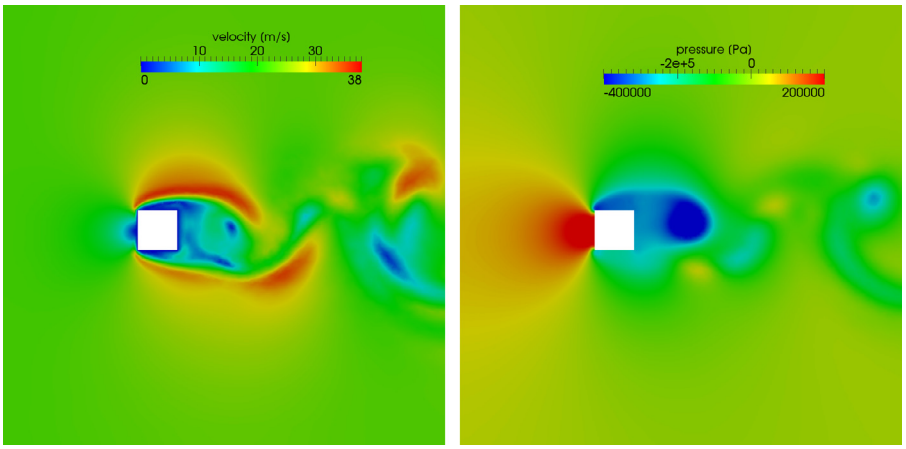
(c) Time to complete a time step.

(d) Number of GMRES iterations.

Fig. 8. Scalability analysis of the solver. Simulations performed using $\mathbb{P}2$—$\mathbb{P}2$ Finite Element, BDF2 and $\Delta t = 0.0025$ s.

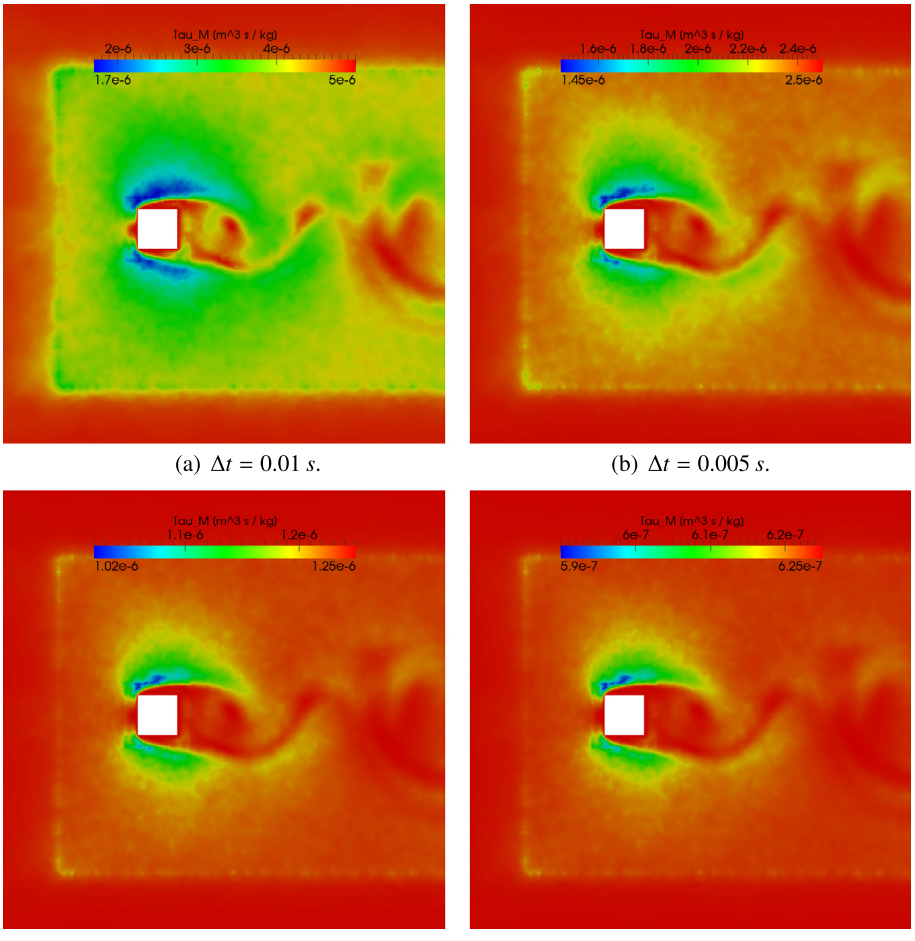
5.4. Stabilization parameters

We address, for the benchmark problem at hand, the dependency of the stabilization parameters $\tau_M^{n+1,\sigma}$ and $\tau_C^{n+1,\sigma}$ of Eqs. (32) and (33) on the choice of the time step Δt. With this aim, we visualize the spatial distributions of $\tau_M^{n+1,\sigma}$ and $\tau_C^{n+1,\sigma}$ on the mid plane of the computational domain starting from the velocity field computed at the time $t = 10.0$ s by means of $\mathbb{P}1$–$\mathbb{P}1$ Finite Element and the BDF2 scheme for $\Delta t = 0.0025$ s; the corresponding computed velocity and pressure fields are reported in Fig. 9, while



(a) Velocity field at time 10.0 *s*. (b) Pressure field at time 10.0 *s*.

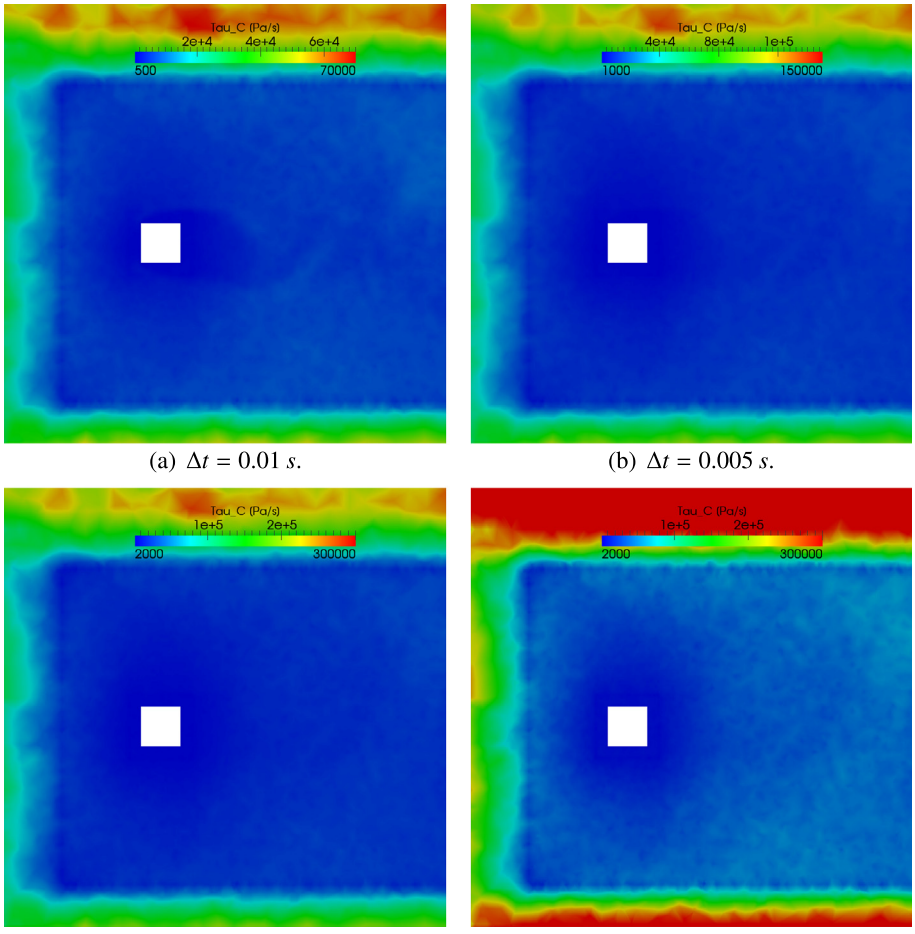
Fig. 9. Reference velocity (a) and pressure (b) fields computed at time $t = 10.0$ s using $\mathbb{P}1$–$\mathbb{P}1$ Finite Element and the BDF2 scheme with $\Delta t = 0.0025$ s; the velocity field is used to evaluate the stabilization parameters $\tau_M^{n+1,\sigma}$ and $\tau_C^{n+1,\sigma}$.



(a) $\Delta t = 0.01$ *s*. (b) $\Delta t = 0.005$ *s*.

(c) $\Delta t = 0.0025$ *s*. (d) $\Delta t = 0.00125$ *s*.

Fig. 10. Spatial distribution (on the mid plane of Ω) of the stabilization parameter $\tau_M^{n+1,\sigma}$ computed from the reference velocity field of Fig. 9(a) using $\mathbb{P}1$–$\mathbb{P}1$ Finite Element and the BDF2 scheme for different time steps $\Delta t = 0.01, 0.005, 0.0025$, and 0.00125 s.



(a) $\Delta t = 0.01\ s.$ (b) $\Delta t = 0.005\ s.$

(c) $\Delta t = 0.0025\ s.$ (d) $\Delta t = 0.00125\ s.$

Fig. 11. Spatial distribution (on the mid plane of Ω) of the stabilization parameter $\tau_C^{n+1,\sigma}$ computed from the reference velocity field of Fig. 9(a) and the parameters $\tau_M^{n+1,\sigma}$ of Fig. 10 using $\mathbb{P}1$–$\mathbb{P}1$ Finite Element and the BDF2 scheme for different time steps $\Delta t = 0.01, 0.005, 0.0025$, and 0.00125 s.

the values of $\tau_M^{n+1,\sigma}$ and $\tau_C^{n+1,\sigma}$ for the time steps $\Delta t = 0.01, 0.005$, 0.0025, and 0.00125 s are reported in Figs. 10 and 11. From Fig. 10(a) we observe that the largest time step under consideration, $\Delta t = 0.01$ s, leads to a distribution of $\tau_M^{n+1,\sigma}$ that reflects the value of the local mesh size h_K and velocity field $\mathbf{u}_{n+1,\sigma}^h$. However, as the time step Δt diminishes, the magnitude of the stabilization parameter $\tau_M^{n+1,\sigma}$ diminishes and its spatial distribution only mildly depends on the local mesh size and the velocity magnitude. In fact, from the definition of $\tau_M^{n+1,\sigma}$ in Eq. (32), we remark that the use of small time steps tends to annihilate the dependence on h_K and $\boldsymbol{u}_{n+1,\sigma}^h$, as it is possible to observe in Fig. 10(d) where $\tau_M^{n+1,\sigma}$ mostly behaves as $\frac{\Delta t}{\rho}$ for $\Delta t \to 0$. We report in Fig. 11 the behavior of the stabilization parameter $\tau_C^{n+1,\sigma}$ computed as in Eq. (33). We notice that, since mild dependencies of $\tau_M^{n+1,\sigma}$ on h_K and $\mathbf{u}_{n+1,\sigma}^h$ are observed for the values of the Δt under considerations, the parameter $\tau_C^{n+1,\sigma}$ mostly depends on the local mesh size h_k; indeed, we have considered a variable mesh size as highlighted in Fig. 2. As consequence, the ratio between the largest and smallest local values of $\tau_C^{n+1,\sigma}$ in Ω is approximatively 150 regardless of the time step Δt under consideration.

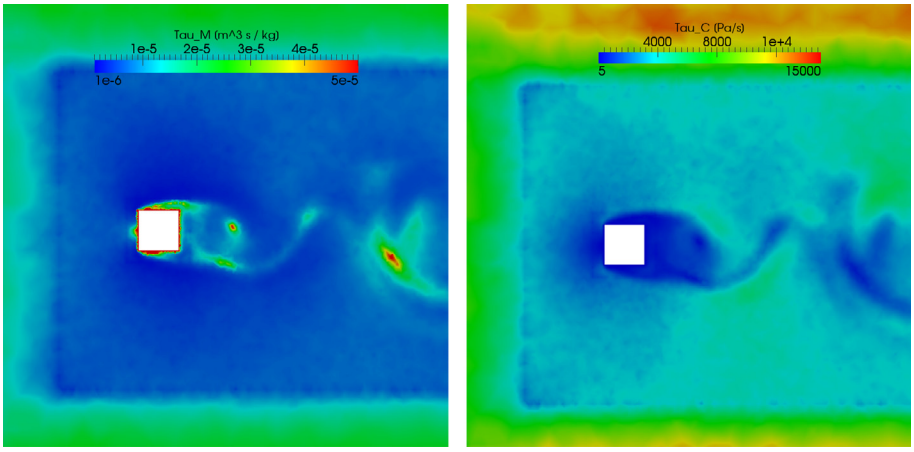
While it is well known that the stabilization parameters chosen as in Eqs. (32) and (33) degenerate as the time step decreases (see e.g. [13,28]), i.e $\tau_M^{n+1,\sigma} \sim \Delta t \to 0$ and $\tau_C^{n+1,\sigma} \sim \frac{1}{\Delta t} \to \infty$ for $\Delta t \to 0$, and hence the stabilization and turbulence LES modeling may be ineffective for relatively small values of Δt ([13]), we stress the fact that we did not encounter any stabilization issue in our numerical simulations for the time steps under consideration. However, our experience based on numerical tests indicates that for very "small" values of Δt, i.e. when the term related to Δt in $\tau_M^{n+1,\sigma}$ is not correctly balanced with respect to the other ones, the degeneration of the parameters $\tau_M^{n+1,\sigma}$ and $\tau_C^{n+1,\sigma}$ is reflected at the numerical level by convergence issues of the linear solver and eventually by the inability of assembling a suitable preconditioner.

An alternative formulation of the stabilization parameters of Eqs. (32) and (33) can be used by completely neglecting the effect of time dependency of the fine scale velocity $\mathbf{u}'$ in the fine scale momentum equation, where one assumes that $\frac{\partial \mathbf{u}'}{\partial t} \simeq 0$. This leads to consider stabilization parameters $\widetilde{\tau}_M^{n+1,\sigma}$ and $\widetilde{\tau}_C^{n+1,\sigma}$ in Eqs. (14) and (15) independent of the time step Δt, which read:

$$\widetilde{\tau}_M^{n+1,\sigma} := \left(\frac{\rho^2}{h_K^2} \left| \boldsymbol{u}_{n+1,\sigma}^h \right|^2 + \frac{\mu^2}{h_K^4} C_r \right)^{-1/2},$$

$$\widetilde{\tau}_C^{n+1,\sigma} = \frac{h_K^2}{\widetilde{\tau}_M^{n+1,\sigma}}.$$

However, we remark that we were unable to successfully perform numerical simulations by using the previous definitions of the stabilization parameters, even for different choices of Δt; specifically, we experienced convergence issues using the GMRES solver with



(a) $\widetilde{\tau}_M^{n+1,\sigma}$. (b) $\widetilde{\tau}_C^{n+1,\sigma}$.

Fig. 12. Spatial distribution (on the mid plane of Ω) of the stabilization parameters $\widetilde{\tau}_M^{n+1,\sigma}$ and $\widetilde{\tau}_C^{n+1,\sigma}$ computed from the reference velocity field of Fig. 9(a) using $\mathbb{P}1$–$\mathbb{P}1$ Finite Element.

the ML preconditioner, which prevented the simulations to significantly advance beyond the time $T_v = 0.3$ s. Our interpretation of this outcome is that a suitable multigrid preconditioner can not be "easily" assembled when the stabilization parameters $\widetilde{\tau}_M^{n+1,\sigma}$ and $\widetilde{\tau}_C^{n+1,\sigma}$ are significantly varying in the computational domain Ω. Following the procedure described before, we plot in Fig. 12 the spatial distributions of the parameters $\widetilde{\tau}_M^{n+1,\sigma}$ and $\widetilde{\tau}_C^{n+1,\sigma}$; in particular, we highlight that $\widetilde{\tau}_M^{n+1,\sigma}$ varies of approximately two orders of magnitude in Ω, while the ratio between the largest and smallest local values of $\widetilde{\tau}_C^{n+1,\sigma}$ is approximatively 3000, a value much larger than the one obtained with $\tau_C^{n+1,\sigma}$, which was about 150. We notice that $\widetilde{\tau}_M^{n+1,\sigma}$ is relatively "large" where the velocity magnitude is nearly zero, as it occurs in the boundary layers and some regions in the vortexes wake. This, in combination with the "small" mesh size h_K in the boundary layers and wake regions, renders the parameter $\widetilde{\tau}_C^{n+1,\sigma}$ relatively "small" locally in Ω; conversely, this becomes "large" far from the cylinder, where the mesh is locally coarse and the velocity magnitude nearly equal to V_∞.

We finally remark that the choice of the stabilization parameters may have a significant impact on the numerical results, especially if, as in this work, we take into account only in an approximate manner for the time dependencies of the fine scales using the parameters $\tau_M^{n+1,\sigma}$ and $\tau_C^{n+1,\sigma}$ of Eqs. (32) and (33). While this can be improved by using dynamic subscales models [11–13], the latter require to solve systems of ordinary differential equations to determine the fine scale solutions $\mathbf{u}'$ and p', which may increase the accuracy of the results, but also the computational cost of the simulations. Since in this work we focus on the efficiency aspects of the numerical solver, we decided to consider the stabilization parameters as in Eqs. (32) and (33), which nevertheless represent a common choice in literature [2,3,15,20,51].

6. Conclusions

In this work we proposed a semi-implicit time discretization of the Navier–Stokes equations with Variational Multiscale-Large Eddy Simulation (VMS-LES) modeling of turbulence. We used Backward Differentiation Formulas (BDF) for the time discretization and the Finite Element method for the spatial approximation, considering the same space of Lagrange polynomials to represent the fluid velocity and pressure fields. We "linearized" the fully discrete nonlinear problem by extrapolation based on Newton–Gregory backward polynomials, thus obtaining a semi-implicit scheme for turbulence LES modeling. In this way, we limit the computational costs of the simulation since only a linear system has to be assembled and solved at each time step. In order to cope with the computational complexity of the problem, we developed a parallel solver based on the GMRES method preconditioned by a multigrid (ML) preconditioner.

We validated the proposed numerical scheme by solving the benchmark problem of the flow past a squared cylinder at high Reynolds number. We investigated how the accuracy of the solution is influenced by the choice of the time step and of the orders of the spatial and time discretizations and, in this respect, we discussed the role of the stabilization parameters. Furthermore, we studied the performance of the solver implemented in a parallel setting, showing the good scalability properties up to thousands of CPUs. All the numerical experiments performed yielded results that were in line with those available in literature obtained by other LES models. This highlights that the proposed semi-implicit BDF time discretization scheme for the Navier–Stokes equations with VMS-LES modeling is not only efficient, but also robust and accurate for the simulation of turbulent flows.

Acknowledgements

The research of D. Forti was supported by the Swiss National Foundation (SNF), project No. 140184. The authors gratefully acknowledge the Swiss National Supercomputing Center (CSCS) for providing the CPU resources for the numerical simulations under project ID s475. Finally, the authors acknowledge Prof. R. Codina, Dr. S. Deparis, Dr. T. Lassila, and Prof. A. Quarteroni for the fruitful discussions, suggestions, and advice.

References

[1] Akkerman I, Bazilevs Y, Calo VM, Hughes TJR, Hulshoff S. The role of continuity in residual-based variational multiscale modeling of turbulence. Comput Mech 2008;41(3):371–8.
[2] Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. Comput Meth Appl Mech Eng 2007;197(1–4):173–201.
[3] Bazilevs Y, Gohean JR, Hughes TJR, Moser RD, Zhang Y. Patient-specific isogeometric fluid-structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. Comput Meth Appl Mech Eng 2009;198(45–46):3534–50.
[4] Bazilevs Y, Takizawa K, Tezduyar TE. Computational fluid–structure interaction: methods and applications. Chirchester, UK: Wiley & Sons; 2013.

[5] Bearman P, Obasaju E. An experimental study of pressure fluctuations on fixed and oscillating square-section cylinders. J Fluid Mech 1981;119:297–321.
[6] Becker R, Rannacher R. An optimal control approach to a posteriori error estimation in finite element methods. Acta Numer 2001;10:1–102.
[7] Brenan KE, Campbell SLV, Petzold LR. Numerical solution of initial-value problems in differential-algebraic equations. New York: North-Holland Publishing Co.; 1989.
[8] Calo VM, Bazilevs Y, Hughes TJR, Moser R. Turbulence modeling for large eddy simulations. Comput Meth Appl Mech Eng 2010;199(13–16):779.
[9] Calvo M, Grigorieff RD. Time discretisation of parabolic problems with the variable 3-step BDF. BIT Numer Math 2002;42(4):689–701.
[10] Cellier FE, Kofmanl E. Continuous system simulation. New York: Springer–Verlag; 2006.
[11] Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. Comput Meth Appl Mech Eng 2002;191:4295–321.
[12] Codina R, Principe J, Guasch O, Badia S. Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. Comput Meth Appl Mech Eng 2007;196:2413–30.
[13] Colomés O, Badia S, Codina R, Principe J. Assessment of variational multiscale models for the large eddy simulation of turbulent incompressible flows. Comput Meth Appl Mech Eng 2015;285:32–63.
[14] Dedè L. Optimal flow control for Navier–Stokes equations: drag minimization. Int J Numer Meth Fluids 2007;55:347–66.
[15] Gamnitzer P, Gravemeier V, Wall WA. Time-dependent subgrid scales in residual-based large eddy simulation of turbulent channel flow. Comput Meth Appl Mech Eng 2010;199:819–27.
[16] Gear CW. Simultaneous numerical solution of differential-algebraic equations. IEEE Trans Circ Theory 1971;18(1):89–95.
[17] Gee MW, Siefer CM, Hu JJ, Tuminaro RS. ML 5.0 smoothed aggregation user's guide. Sandia National Laboratories; 2006.
[18] Germano M, Piomelli U, Moin P, Cabot WH. A dynamic subgrid-scale eddy viscosity model. Phys Fluids A 1991;3(7):1760–5.
[19] Gervasio P, Saleri F, Veneziani A. Algebraic fractional-step schemes with spectral methods for the incompressible Navier–Stokes equations. J Comput Phys 2006;214(1):347–65.
[20] Gravemeier V, Gee MW, Kronbichler M, Wall WA. An algebraic variational multiscale-multigrid method for large eddy simulation of turbulent flow. Comput Meth Appl Mech Eng 2010;199:853–64.
[21] Jansen KE, Whiting CH, Hulbert GM. A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. Comput Meth Appl Mech Eng 1999;190:305–19.
[22] Jansson N, Hoffman J, Jansson J. Framework for massively parallel adaptive finite element computational fluid dynamics on tetrahedral meshes. SIAM J Sci Comput 2012;34:C24–41.
[23] Jeong J, Hussain F. On the identification of a vortex. J Fluid Mech 1995;285:69–94.
[24] Kim J, Moin P, Moser R. Turbulence statistics in fully developed channel flow at low Reynolds number. J Fluid Mech 1987;177:133–66.
[25] Koobus B, Farhat C. A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes—application to vortex shedding. Comput Meth Appl Mech Eng 2003;193(15–16):1367–83.
[26] Koobus B, Tran H, Farhat C. Computation of unsteady viscous flows around moving bodies using the k-ϵ turbulence model on unstructured dynamic grids. Comput Meth Appl Mech Eng 2000;190:1441–66.
[27] Hoffman J. Computation of mean drag for bluff body problems using adaptive DNS/LES. SIAM J Sci Comput 2006;27:184–207.
[28] Hsu MC, Bazilevs Y, Calo VM, Tezduyar TE, Hughes TJR. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. Comput Meth Appl Mech Eng 2010;199:828–40.
[29] Hughes TJR. Multiscale phenomena: Greens functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods. Comput Meth Appl Mech Eng 1995;127:387–401.
[30] Hughes TJR, Calo VM, Scovazzi G. Variational and multiscale methods in turbulence. In: Gutkowski W, Kowalewski TA, editors. Mechanics of the 21st century. Amsterdam: Springer; 2005. p. 153–63.
[31] Hughes TJR, Mazzei L, Jansen KE. Large-eddy simulation and the variational multiscale method. Comput Vis Sci 2000;3:47–59.
[32] Hughes TJR, Mazzei L, Oberai AA, Wray AA. The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. Phys Fluids 2001;13(2):505–12.
[33] Hughes TJR, Oberai AA, Mazzei L. Large eddy simulation of turbulent channel flows by the variational multiscale method. Phys Fluids 2001;13(6):1784–99.
[34] Hughes TJR, Scovazzi G, Franca LP. Multiscale and stabilized methods. In: Stein E, de Borst R, Hughes TJR, editors. Encyclopedia of computational mechanics. John Wiley & Sons; 2004.
[35] Lilly DK. A proposed modification of the Germano subgridscale closure method. Phys Fluids A 1992;4(3):633–5.
[36] Lyn D, Einav S, Rodi W, Park J. A laser-doppler velocimeter study of ensemble-averaged characteristics of the turbulent near wake of a square cylinder. J Fluid Mech 1995;304:285–319.
[37] Masud A, Calderer R. A variational multiscale stabilized formulation for the incompressible Navier–Stokes equations. Comput Mech 2009;44(2):145–60.
[38] Moser RD, Kim J, Mansour NN. Direct numerical simulations of turbulent channel flow up to $\mathbb{R}e_\tau = 590$. Phys Fluids 1999;11:943–5.
[39] Nicoud F, Ducros F. Subgrid-scale modelling based on the square of the velocity gradient tensor. Flow Turbul Combust 1999;62:183–200.
[40] Pope SB. Turbulent flows. Cambridge University Press; 2000.
[41] Quarteroni A, Saleri F, Sacco R. Numerical mathematics. Berlin: Springer; 2007.
[42] Quarteroni A, Valli A. Domain decomposition methods for partial differential equations. New York: Oxford University Press; 1999.
[43] Quarteroni A, Valli A. Numerical approximation of partial differential equations. Berlin and He idelberg: Springer-Verlag; 1994.
[44] Rao GS. Numerical analysis. New Delhi: New Age International Publishers; 2009.
[45] Rodi W, Ferziger J, Breuer M, Pourquié M. Status of large eddy simulations: results of a workshop. J Fluids Eng 1997;119:248–62.
[46] Ruge J, Stüben K. Algebraic multigrid (AMG). In: McCormick SF, editor. Multigrid methods frontiers in applied mathematics, vol. 3. Philadelphia, PA: SIAM; 1987. p. 73–130.
[47] Sagaut P. Large eddy simulation for incompressible flows. 3rd ed. Berlin: Springer-Verlag; 2006.
[48] Shakib F, Hughes TJR, Johan Z. A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations. Comput Meth Appl Mech Eng 1991;89(1–3):141–219.
[49] Smagorinsky JS. General circulation experiments with the primitive equations. Mon Weather Rev 1963;91:99–164.
[50] Sohankar A, Davidson L, Norberg C. Large eddy simulation of flow past a square cylinder: comparison of different subgrid scale models. J Fluids Eng 2000;122:39–47.
[51] Tezduyar TE, Sathe S. Stabilization parameters in SUPG and PSPG formulations. J Comput Appl Mech 2003;4(1):71–88.
[52] Toselli A, Widlund O. Domain decomposition methods – algorithms and theory. Berlin: Springer-Verlag; 2005.
[53] Villa U. Scalable efficient methods for incompressible fluid-dynamics in engineering problems. PhD thesis, Emory University; 2012.
[54] Zistl C, Hilbert R, Janiga G, Thévenin D. Increasing the efficiency of post-processing for turbulent reacting flows. Comput Vis Sci 2009;12(8):383–95.
[55] LifeV website: <www.lifev.org>.