Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 333 (2018) 371–394

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Efficient and scalable discretization of the Navier–Stokes equations with LPS modeling

Ryadh Haferssas[a], Pierre Jolivet[b,*], Samuele Rubino[c]

[a] *Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[b] *CNRS, IRIT–ENSEEIHT, 2 rue Charles Camichel, 31071 Toulouse Cedex 7, France*
[c] *Departamento EDAN & IMUS, Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain*

Received 6 October 2017; received in revised form 9 January 2018; accepted 17 January 2018
Available online 31 January 2018

Abstract

In this work, we address the solution of the Navier–Stokes equations (NSE) by a Finite Element (FE) Local Projection Stabilization (LPS) method. The focus is on a LPS method that has one level, in the sense that it is defined on a single mesh, and in which the projection-stabilized structure of standard LPS methods is replaced by an interpolation-stabilized structure, which only acts on the high frequency components of the flow. As a main contribution, we propose and test an efficient discretization of the model via a stable velocity–pressure segregation, using semi-implicit Backward Differentiation Formulas (BDF) in time. On the one hand, numerical studies illustrate that the solver accurately reproduces first and second-order statistics of benchmark turbulent flows for relatively coarse meshes. On the other hand, they show that the solver works in an efficient (i.e., robust and fast) way, especially when interfaced with scalable domain decomposition methods. Such scalability results are obtained on up to 16,384 cores with a near-ideal speedup.

MSC: 65M55; 65M60; 76D05

Keywords: Navier–Stokes equations; LPS by interpolation; Pressure-correction methods; Large eddy simulation; Turbulent incompressible flows; Domain decomposition

1. Introduction

In the present paper, we propose an efficient space–time discretization of the Navier–Stokes Equations (NSE) for the simulation of laminar and turbulent incompressible flows, with a special emphasis on their numerical solution in a parallel setting. Turbulent flows occur in many physical contexts (e.g., external flows in civil engineering, hydrodynamics, aeronautical applications; internal flows in hemodynamics).

The major problem in treating turbulent flows is due to the wide range of scales involved. For laminar flows already, a substantial range of scales may be encountered. Dealing with turbulent flows, however, implies an even

[*] Corresponding author.
E-mail addresses: rhaferss@mit.edu (R. Haferssas), pierre.jolivet@enseeiht.fr (P. Jolivet), samuele@us.es (S. Rubino).

https://doi.org/10.1016/j.cma.2018.01.026

broader range of scales in comparison to laminar flows, which are also in nonlinear interactions with each other. This makes their accurate and efficient simulation a really challenging task, even in a High Performance Computing (HPC) framework. Indeed, Direct Numerical Simulation (DNS) demands high computational effort (beyond the limits of the currently available computer power in most cases) to accurately solve with extremely fine grids the broad range of scales involved.

Conversely, Large Eddy Simulation (LES) [1] is an intermediate approach in its requirement of computational effort and degree of modeling. The strategy of LES consists in solving the largest flow structures and modeling the effect of the smallest flow structures on the largest ones. The traditional LES model relies on a filter to separate resolved and unresolved scales at the continuous level. On the one hand, a coarser discretization, which is substantially coarser than a comparable DNS discretization, is sufficient for resolving the large scales and, on the other hand, the universal character of the statistical behavior for the small scales justifies the modeling process, cf. [2]. These models are based on a physical approach defined considering the physical phenomena that take place on the smallest scales. Nevertheless, another way to model the fine scales in a LES method can be developed by a purely numerical approach that does not introduce any modification of the governing equations at the continuous level. This last numerical approach, which hence relies on purely numerical artifacts without any modification of the continuous problem, was seldom followed, the MILES (Monotone Integrated LES) approach [3] being the main exception, until the residual-based Variational MultiScale (VMS) models were introduced in the seminal papers [4,5] and subsequently proposed as implicit LES techniques (ILES) for turbulent flows in [6,7]. These models provide a unified framework for the definition of spatial approximation schemes capable of preventing numerical instabilities that arise when the standard Galerkin Finite Element (FE) method is used, and adequate to represent the turbulence LES modeling. The basic concept consists in differentiating two scale groups (resolved and unresolved scales) using a residual-based model of the unresolved scales to account for their influence into the resolved ones. In contrast to the use of a filter in the aforementioned traditional LES, a variational projection between function spaces separates scale ranges within the VMS method. Thus, residual-based VMS are intrinsically discrete models, and no approximation of an intermediate averaged model is needed. The residual-based VMS procedure does not make use of the statistical theory of equilibrium turbulence, and thus no ad-hoc eddy viscosity is required. Furthermore, it strongly retains numerical consistency. However, the subgrid terms have a rather complex structure, since they involve the full residual with convective interactions between resolved and unresolved scales, thus increasing computational complexity and setting serious numerical difficulty just to prove stability.

In this work, we focus on an alternative strategy starting from a high-order term-by-term stabilization, cf. [8], which does not involve the full residual, and presents a simpler and less expensive structure for practical implementations. This method is a particular type of Local Projection Stabilization (LPS) scheme that may be cast in the VMS framework [9], and constitutes a low-cost, accurate solver (of optimal order) for incompressible flows, despite being only weakly consistent. It presents the same structure of the Streamline Derivative-based (SD-based) LPS model [10,11], but it differs from it because at the same time it uses continuous buffer functions, it does not need enriched FE spaces, it does not need element-wise projections satisfying suitable orthogonality properties, and it does not need different nested meshes. An interpolant-stabilized structure of Scott–Zhang type replaces the projection-stabilized structure of standard LPS methods. The interpolation operator takes its values in a continuous buffer space, different from the discrete velocity space, but defined on the same mesh, constituted by standard polynomials with one degree less than the FE space for the velocity. This approach gives rise to a method with reduced computational cost for some choices of the interpolation operator. This method has been recently supported by a thorough numerical analysis (existence and uniqueness, stability, convergence, error estimates, asymptotic energy balance) for the nonlinear problem related to the evolution NSE, cf. [12,13], using a semi-implicit Euler scheme for the monolithic discretization in time. In particular, the error analysis reveals a self-adapting high spatial accuracy in laminar regions of a turbulent flow that turns to be of overall optimal high accuracy if the flow is fully laminar. Numerical simulations of 3D Beltrami flow in laminar regimes [12] confirm this fact. This also allows to obtain an asymptotic energy balance for smooth flows.

The main contribution of this work is to propose an efficient space–time discretization of the incompressible NSE with LPS modeling of subgrid interactions by aiming at the simulation of laminar and, especially, turbulent flows in a parallel framework. In particular, the problem is first discretized in space by using a FE LPS by interpolation, and then in time by a two-step pressure-correction projection algorithm based on semi-implicit Backward Differentiation Formulas (BDF) [14]. Since we aim at solving large scale problems at high Reynolds numbers, the use of parallel architectures is necessary. In such parallel framework, in addition to suitable choices of the spatial and time discretization schemes, we stress the fact that the use of domain decomposition methods and efficient linear solvers and preconditioning strategies is often mandatory to make the numerical simulations computationally feasible.

In this work, we interface the proposed fully discrete scheme for the NSE with LPS modeling with HPDDM [15], a high performance unified framework for domain decomposition methods. In particular, we use a parallel iterative linear solver based on an optimized Schwarz domain decomposition method as preconditioner [15,16]. In this manner, we obtain an efficient, i.e., robust and fast, solver for the HPC of laminar and turbulent flows in the open-source FE software FreeFem++ [17] interfaced with HPDDM. A similar recent study has been performed in [18], where a semi-implicit BDF time discretization scheme for the NSE with residual-based VMS–LES modeling is combined together with a parallel multigrid preconditioner applied on the right and the GMRES iterative method. However, the cited study differs also because it is applied to the monolithic (coupled velocity–pressure) form of the linear system associated to the problem. Other similar studies can be found in the literature, but they always differ in some aspects with respect to the present work. In [19], a balancing Neumann–Neumann domain decomposition method is used for preconditioning the GMRES iterative method applied to the fully implicit monolithic system associated to a residual-based Orthogonal Subscales (OSS) modeling of the NSE. In [20], algebraic multigrid strategies for VMS–LES modeling are discussed, where a Smagorinsky-type eddy viscosity model is required. In our parallel framework, we aim at further reducing the computational cost by using a semi-implicit fractional-step (FS) projection algorithm, and the computational complexity by using a LPS modeling, which is not fully residual-based, but allows similarly to obtain a high-order accuracy without requiring any ad-hoc eddy viscosity. The parallel framework set aside, a fully discrete LPS method that uses a pressure-correction scheme based on semi-implicit BDF and inf–sup stable FE has been analyzed in [21]. Here, we aim to test the practical performances of a similar method based on inf–sup stable FE in conjunction with grad–div and streamline derivative-based LPS in our parallel framework, similarly to the numerical investigation performed in [22] for a monolithic implicit time discretization with a recursive block preconditioning.

The proposed fully discrete scheme is tested towards the benchmark problem of recirculating flow in a lid-driven cavity. Firstly, we validate the proposed numerical scheme for the two-dimensional lid-driven cavity flow at high Reynolds number (up to $Re = 10{,}000$). Then, we address simulations of the three-dimensional lid-driven cavity flow, for which there exist experimental measurements and numerical results in the literature, by considering three significant Reynolds numbers ($Re = 3200$, 7500, and 10,000) to cover the spectrum from laminar to turbulent regime. A comparison of first and second-order statistics with experimental data so as to other numerical results justifies the interest of our approach: the proposed method exhibits a high-order accuracy in predicting these sensitive measures already for relatively coarse meshes, and also computational efficiency and strong scalability results of the solver in a HPC framework are showcased.

The outline of the paper is as follows. In Section 2, we introduce the model problem and its continuous variational formulation for time-dependent incompressible NSE. In Section 3, we introduce the fully discrete problem. We initially describe the proposed LPS spatial approximation of the incompressible evolution NSE, and we state its main properties. The rest of the section is devoted to the time discretization of the LPS model by means of an incremental pressure-correction algorithm with semi-implicit BDF. In Section 4, we describe the parallel solver developed for the fully discrete problem, as well as the preconditioning technique used in the framework of HPDDM. The proposed strategy is tested for the recirculating flow in a lid-driven cavity in Section 5. Firstly, we report and discuss the numerical results for the two-dimensional case at high Reynolds number. Then, we show the potential of the proposed method for simulating turbulent recirculating flow in a three-dimensional lid-driven cavity, pointing out also the parallel performances of the solver. Section 6 states the main conclusions of the paper.

2. Time-dependent incompressible Navier–Stokes equations: problem statement and variational formulation

We introduce an Initial–Boundary Value Problem (IBVP) for the incompressible evolution NSE. For the sake of simplicity, we just impose homogeneous Dirichlet boundary condition on the whole boundary.

Let $[0, T]$ be the time interval, and Ω a bounded polyhedral domain in $\mathbb{R}^d$, $d = 2$ or 3, with a Lipschitz-continuous boundary $\Gamma = \partial\Omega$. The transient NSE in strong form for an incompressible fluid are given by:

find $\boldsymbol{u} : \Omega \times (0,T) \longrightarrow \mathbb{R}^d$ *and* $p : \Omega \times (0,T) \longrightarrow \mathbb{R}$ *such that:*

$$\begin{cases} \partial_t \boldsymbol{u} + \nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{u}) - 2\nu\nabla\cdot D(\boldsymbol{u}) + \nabla p = \boldsymbol{f} & \text{in } \Omega\times(0,T),\\ \nabla\cdot\boldsymbol{u} = 0 & \text{in } \Omega\times(0,T),\\ \boldsymbol{u} = \boldsymbol{0} & \text{on } \Gamma\times(0,T),\\ \boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0(\boldsymbol{x}) & \text{in } \Omega, \end{cases} \tag{2.1}$$

where $\boldsymbol{u}\otimes\boldsymbol{u}$ is the tensor function of components $u_i u_j$, and $D(\boldsymbol{u})$ is the symmetric deformation tensor given by $D(\boldsymbol{u}) = (1/2)(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^{\mathrm{T}})$. The unknowns are the velocity $\boldsymbol{u}$ and the pressure p of the incompressible fluid. The data are the source term $\boldsymbol{f}$, which represents a body force per mass unit (typically the gravity), the kinematic viscosity ν of the fluid, which is a positive constant, and the initial velocity $\boldsymbol{u}_0$.

To define the weak formulation of problem (2.1), we need to introduce some useful notations for spaces. We consider the Sobolev spaces $H^s(\Omega)$, $s \in \mathbb{R}$, and $L^p(\Omega)$, $1 \leqslant p \leqslant \infty$. We use the following notation for vector-valued Sobolev spaces: $\mathbf{H}^s$ and $\mathbf{L}^p$ respectively shall denote $[H^s(\Omega)]^d$ and $[L^p(\Omega)]^d$ (similarly for tensor spaces of dimension $d\times d$). Also, the parabolic Bochner function space $L^p(0,T;X)$ (resp. $L^p(0,T;\mathbf{X})$), where X (resp. $\mathbf{X}$) stands for a scalar (resp. vector-valued) Sobolev space shall be denoted by $L^p(X)$ (resp. $L^p(\mathbf{X})$). In order to give a variational formulation of problem (2.1), let us consider the velocity space:

$$\mathbf{H}^1_0 = [H^1_0(\Omega)]^d = \left\{\boldsymbol{w} \in [H^1(\Omega)]^d : \boldsymbol{w} = \boldsymbol{0} \text{ on } \Gamma\right\}.$$

This is a closed linear subspace of $\mathbf{H}^1$, and thus a Hilbert space endowed with the $\mathbf{H}^1$-norm. Thanks to Korn's inequality, cf. [23], the $\mathbf{H}^1$-norm is equivalent on $\mathbf{H}^1_0$ to the norm $\|\boldsymbol{w}\|_{\mathbf{H}^1_0} = \|D(\boldsymbol{w})\|_{\mathbf{L}^2}$. Also, let us introduce the space of divergence-free functions:

$$\mathbf{H}^1_{0,\mathrm{div}} = \left\{\boldsymbol{w} \in \mathbf{H}^1_0 : \nabla\cdot\boldsymbol{w} = 0 \text{ a.e. in } \Omega\right\}.$$

The space $\mathbf{H}^1_{0,\mathrm{div}}$ is a closed linear subspace of $\mathbf{H}^1_0$, and thus a Hilbert space endowed with the $\mathbf{H}^1$-norm. We shall consider the following variational formulation of (2.1):

given $\boldsymbol{f} \in L^2(\mathbf{H}^{-1})$ *and* $\boldsymbol{u}_0 \in \mathbf{H}^{-1}$, *find* $\boldsymbol{u} \in L^\infty(\mathbf{L}^2) \cap L^2(\mathbf{H}^1_{0,\mathrm{div}})$, $P \in L^2(L^2_0)$ *such that:*

$$\begin{cases} -\displaystyle\int_0^T (\boldsymbol{u}(t),\boldsymbol{v})_\Omega \varphi'(t)\,dt - \langle \boldsymbol{u}_0,\boldsymbol{v}\rangle\varphi(0)\\ +\displaystyle\int_0^T [b(\boldsymbol{u}(t),\boldsymbol{u}(t),\boldsymbol{v}) + a(\boldsymbol{u}(t),\boldsymbol{v})]\varphi(t)\,dt\\ +\displaystyle\int_0^T (P(t),\nabla\cdot\boldsymbol{v})_\Omega\varphi'(t)\,dt = \int_0^T \langle \boldsymbol{f}(t),\boldsymbol{v}\rangle\varphi(t)\,dt, \end{cases} \tag{2.2}$$

for any $\boldsymbol{v} \in \mathbf{H}^1_0$, $\varphi \in \mathcal{D}([0,T])$ such that $\varphi(T) = 0$, $\langle\cdot,\cdot\rangle$ stands for the duality pairing between $\mathbf{H}^1_0$ and its dual $\mathbf{H}^{-1}$, and L^2_0 consists of L^2-functions with zero mean in Ω. The forms b and a are given by:

$$b(\boldsymbol{w},\boldsymbol{u},\boldsymbol{v}) = \frac{1}{2}\left[(\boldsymbol{w}\cdot\nabla\,\boldsymbol{u},\boldsymbol{v})_\Omega - (\boldsymbol{w}\cdot\nabla\,\boldsymbol{v},\boldsymbol{u})_\Omega\right], \tag{2.3}$$

$$a(\boldsymbol{u},\boldsymbol{v}) = 2\nu\,(D(\boldsymbol{u}),D(\boldsymbol{v}))_\Omega, \tag{2.4}$$

for $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w} \in \mathbf{H}^1_0$. The skew-symmetric form of the convective term b is chosen for conservation purposes: note that $b(\boldsymbol{w},\boldsymbol{v},\boldsymbol{v}) = 0$ for all $\boldsymbol{w},\boldsymbol{v} \in \mathbf{H}^1_0$. The physical pressure is the time derivative of the unknown $P : p = \partial_t P \in H^{-1}(L^2_0) = H^1_0(0,T;L^2_0)'$. The interest of considering P as unknown instead of p is that there are high technical difficulties to obtain uniform bounds for the discrete pressures in a Banach space of space–time functions, see [24, remark 10.2], while one obtains uniform bounds in the Banach space $L^\infty(L^2)$ for the numerical approximation of P, see [12, theorem 4.3]. It is known that for domains which satisfy the cone condition, as bounded polyhedral domains, $P \in L^\infty(L^2)$, e.g., see [25, remark 2.5]. We notice, however, that for practical computations one would approximate the physical pressure p, and P is introduced just for the numerical analysis.

3. Fully discrete problem

In this section, we describe both the spatial approximation and the time discretization proposed for the model problem.

3.1. Space approximation: a finite element local projection stabilization model

We consider a FE LPS method applied to the weak form of the NSE (2.2). LPS schemes were originally proposed for the Stokes problem [26], and then successfully extended to transport problems [27–32]. As classical stabilization procedures, these numerical discretizations are based upon an "augmented" variational formulation of the flow equations, which includes additional terms to the standard Galerkin discretization to provide specific stabilization of any single operator term that could be a source of instability. In particular, LPS schemes allow to circumvent the standard discrete inf–sup condition and to use equal order interpolation for velocity and pressure, and they also provide local stabilization of convection-dominant effects and improvement of local mass conservation [33]. Different variants of LPS methods have been investigated during the recent years for incompressible flow problems. The main common feature is that, thanks to local projection, the symmetric stabilization terms only act on the small scales of the flow, thus ensuring a higher accuracy with respect to more classical stabilization procedures, such as penalty-stabilized methods, cf. [34]. This also guarantees a self-adapting high accuracy in laminar regions of a turbulent flow, which turns out to be of overall optimal high accuracy if the flow is fully laminar, and allows to obtain an asymptotic energy balance for smooth flows [12]. Thus, the effect of LPS is on the one hand to improve the convergence to smooth solutions. On the other hand, for rough solutions, LPS limits the propagation of perturbations generated in the vicinity of sharp gradients, potentially maintaining these schemes as suitable and useful tools for the simulation of turbulent flows. Moreover, an important advantage of their term-by-term structure is that the projection can be easily treated as implicit, without having all the residual terms coupled, as for more complex residual-based VMS methods, cf. [6,35]. Actually, LPS schemes can be viewed as simplifications of residual-based VMS methods [9], since they are not fully consistent (only specific dissipative interactions are retained), but are of optimal order with respect to the FE interpolation. For a detailed description of different variants of LPS schemes, we refer to [11,36,37].

In order to describe in detail the spatial approximation of the model problem (2.2), let $\{\mathcal{T}_h\}_{h>0}$ be a family of affine-equivalent, conforming (i.e., without hanging nodes) and regular triangulations of $\overline{\Omega}$, formed by triangles or quadrilaterals ($d = 2$), tetrahedra or hexahedra ($d = 3$). For any mesh cell $K \in \mathcal{T}_h$, its diameter will be denoted by h_K and $h = \max_{K\in\mathcal{T}_h} h_K$.

Given a positive integer l and a mesh cell $K \in \mathcal{T}_h$, denote by $\mathbb{R}_l(K)$ either $\mathbb{P}_l(K)$ (i.e., the space of Lagrange polynomials of degree $\leqslant l$, defined on K), if the grids are formed by triangles ($d = 2$) or tetrahedra ($d = 3$), or $\mathbb{Q}_l(K)$ (i.e., the space of Lagrange polynomials of degree $\leqslant l$ on each variable, defined on K), if the family of triangulations is formed by quadrilaterals ($d = 2$) or hexahedra ($d = 3$). We consider the following FE spaces for the velocity:

$$\left\{\begin{array}{l} Y_h^l = V_h^l(\Omega) = \{v_h \in C^0(\overline{\Omega}) : v_h|_K \in \mathbb{R}_l(K),\ \forall K \in \mathcal{T}_h\}, \\ \mathbf{Y}_h^l = [Y_h^l]^d = \{\boldsymbol{v}_h \in [C^0(\overline{\Omega})]^d : \boldsymbol{v}_h|_K \in [\mathbb{R}_l(K)]^d,\ \forall K \in \mathcal{T}_h\}, \\ \mathbf{X}_h = \mathbf{Y}_h^l \cap \mathbf{H}_0^1. \end{array}\right. \tag{3.1}$$

Hereafter, $\mathbf{Y}_h^l$ (resp., Y_h^l) will constitute the discrete foreground vector-valued (resp. scalar) spaces in which we will work on.

We initially approximate the weak formulation (2.2) of the initial–boundary value problem (2.1) for the incompressible evolution NSE by a high-order term-by-term stabilization procedure in space [8]. The stabilization effect is achieved by adding least-squares terms that give a weighted control on the fluctuations of the quantity of interest, based upon a specific locally stable projection or interpolation operator on a continuous buffer space. This provides an efficient discretization with a reduced computational cost that keeps the same high-order accuracy with respect to standard projection-stabilized methods. We initially state the proposed LPS discretization as:

find $(\boldsymbol{u}_h, p_h) : (0, T) \to \mathbf{X}_h \times \mathbb{M}_h$ *such that:*

$$\left\{\begin{array}{r} (\partial_t \boldsymbol{u}_h, \boldsymbol{v}_h)_\Omega + b(\boldsymbol{u}_h, \boldsymbol{u}_h, \boldsymbol{v}_h) + a(\boldsymbol{u}_h, \boldsymbol{v}_h) - (p_h, \nabla\cdot\boldsymbol{v}_h)_\Omega \qquad\qquad \\ + s_{\text{conv}}(\boldsymbol{u}_h, \boldsymbol{u}_h, \boldsymbol{v}_h) + s_{\text{div}}(\boldsymbol{u}_h, \boldsymbol{v}_h) = \langle \boldsymbol{f}, \boldsymbol{v}_h\rangle, \\ (\nabla\cdot\boldsymbol{u}_h, q_h)_\Omega + s_{\text{pres}}(p_h, q_h) = 0, \end{array}\right. \tag{3.2}$$

for any $(\boldsymbol{v}_h, q_h) \in \mathbf{X}_h \times \mathbb{M}_h$, where $\mathbb{M}_h = Y_h^l \cap L_0^2$.

The forms s_{conv}, s_{div} and s_{pres} in (3.2) correspond to a high-order term-by-term stabilized method, cf. [8], and are given by:

$$s_{\text{conv}}(\boldsymbol{u}_h, \boldsymbol{w}_h, \boldsymbol{v}_h) = \sum_{K\in\mathcal{T}_h} \tau_{m,K}(\sigma_h^*(\boldsymbol{u}_h\cdot\nabla\boldsymbol{w}_h), \sigma_h^*(\boldsymbol{u}_h\cdot\nabla\boldsymbol{v}_h))_K, \tag{3.3}$$

$$s_{\mathrm{div}}(\boldsymbol{u}_h,\boldsymbol{v}_h)=\sum_{K\in\mathcal{T}_h}\tau_{d,K}(\sigma_h^*(\nabla\cdot\boldsymbol{u}_h),\sigma_h^*(\nabla\cdot\boldsymbol{v}_h))_K, \tag{3.4}$$

$$s_{\mathrm{pres}}(p_h,q_h)=\sum_{K\in\mathcal{T}_h}\tau_{m,K}(\sigma_h^*(\nabla p_h),\sigma_h^*(\nabla q_h))_K. \tag{3.5}$$

The aims of the stabilization terms is to prevent spurious instabilities due to dominant convection (s_{conv}), achieve additional control on the incompressibility condition by means of pressure subscales effect (s_{div}), and ensure inf–sup stability when using equal order interpolation for velocity and pressure (s_{pres}). Here, $\tau_{m,K}$, $\tau_{d,K}$ are stabilization coefficients for convection–pressure gradient and divergence, respectively, and $\sigma_h^*=Id-\sigma_h$, where Id is the identity operator, and σ_h is some locally stable projection or interpolation operator from $\mathbf{L}^2$ on the foreground vector-valued space $\mathbf{Y}_h^{l-1}$ (also called "buffer space" in this context), satisfying optimal error estimates. In practical implementations, we choose σ_h as a Scott–Zhang-like [38] linear interpolation operator in the space $\mathbf{Y}_h^{l-1}$, implemented in the software FreeFem++ [17]. This interpolant may be defined as:

$$\forall\boldsymbol{x}\in\overline{\Omega},\quad \sigma_h(\boldsymbol{v})(\boldsymbol{x})=\sum_{a\in\mathcal{N}}\Pi_h(\boldsymbol{v})(a)\boldsymbol{\varphi}_a(\boldsymbol{x}),$$

where $\mathcal{N}$ is the set of Lagrange interpolation nodes of $\mathbf{Y}_h^{l-1}$, $\boldsymbol{\varphi}_a$ are the Lagrange basis functions associated to $\mathcal{N}$, and Π_h is the interpolation operator by local averaging of Scott–Zhang kind, which coincides with the standard nodal Lagrange interpolant when acting on continuous functions (cf. [8], section 4). This is an interpolant that just uses nodal values, and so is simpler to work out and more computationally efficient than the variant of the Scott–Zhang operator introduced in [39] for the Stokes problem, which is instead an operator defined from a node-to-element map and requires integration on mesh elements. According to the structure of s_{conv}, the following operators have to be composed:

- bilinear form τ discretized as C:

$$\begin{aligned}\tau: dc\mathbf{Y}_h^{l-1}\times dc\mathbf{Y}_h^{l-1}&\to\mathbb{R}\\ (\boldsymbol{u}_h,\boldsymbol{v}_h)&\mapsto\sum_{K\in\mathcal{T}_h}\tau_{m,K}(\boldsymbol{u}_h,\boldsymbol{v}_h)_K,\end{aligned}$$

- linear operator σ discretized as P:

$$\begin{aligned}\sigma: dc\mathbf{Y}_h^{l-1}&\to dc\mathbf{Y}_h^{l-1}\\ \boldsymbol{v}_h&\mapsto\sigma_h^*(\boldsymbol{v}_h),\end{aligned}$$

- linear operator $\mathcal{D}$ discretized as D:

$$\begin{aligned}\mathcal{D}:\mathbf{X}_h&\to dc\mathbf{Y}_h^{l-1}\\ \boldsymbol{v}_h&\mapsto\boldsymbol{u}_h\cdot\nabla\boldsymbol{v}_h,\end{aligned}$$

where $dc\mathbf{Y}_h^{l-1}$ stands for the discontinuous version of the buffer space $\mathbf{Y}_h^{l-1}$. In practice, we express D as the sum of two (resp. three) operators in 2D (resp. 3D). This is done by considering $\boldsymbol{v}_h$ component-by-component, i.e., $D=D_x+D_y$ (resp. $D=D_x+D_y+D_z$). The discretization of (3.3) is thus the symmetric local matrix S defined as:

$$S=(P\cdot D)^TC(P\cdot D). \tag{3.6}$$

In formula (3.4), σ_h denotes an operator between the scalar spaces L^2 and Y_h^{l-1}, but we use the same notation for the sake of simplicity. Actually, if needed, specific stabilizations for convection, divergence and pressure gradient may be used, through different approximation operators. When using inf–sup stable FE (as in the case of the computations performed in this work), there is no need to consider the pressure stabilization term s_{pres}, so this term is neglected in this case. However, the term s_{div} seems to be crucial in this case, due to the poor resolution of the pressure typical of mixed interpolations that satisfy the inf–sup condition [40]. For this term, just a pure grad–div penalty stabilization is used in practical implementations, i.e., $\sigma_h^*=Id$ in formula (3.4), to reduce the computational cost. However, it can be seen from numerical analysis that this approximation does not introduce any consistency error, that is it does not affect the optimal accuracy of the method [12]. The working expressions of the stabilization coefficients are:

$$\tau_{m,K} = \left[d\, c_1 \frac{\nu}{(h_K/l)^2} + c_2 \frac{U_K^n}{(h_K/l)} \right]^{-1}, \tag{3.7}$$

$$\tau_{d,K} = \frac{(h_K/l)^2}{d\, c_1 \tau_{m,K}}, \tag{3.8}$$

by following the form proposed in [35,41], designed by asymptotic scaling arguments applied in the framework of stabilized methods aimed at taking into account the local balance between convection and diffusion. In expressions (3.7) and (3.8), d is the dimension of the problem, c_1 and c_2 are user-chosen positive constants, l is the polynomial degree of the velocity FE approximation, and U_K is some local speed on the mesh cell K. The values of the constants c_1 and c_2 are chosen to be $c_1 = 4$, $c_2 = \sqrt{c_1} = 2$, cf. [42]. If $U_K \in L^\infty(K)$, the following technical hypothesis on the stabilization coefficients required to perform the numerical analysis is ensured:

Hypothesis 3.1. The stabilization coefficients $\tau_{m,K}$, $\tau_{d,K}$ satisfy the following conditions:

$$\alpha_1 h_K^2 \leqslant \tau_{m,K} \leqslant \alpha_2 h_K^2, \quad 0 < \tau_{d,K} \leqslant \beta, \tag{3.9}$$

for all $K \in \mathcal{T}_h$, and some positive constants $\alpha_1, \alpha_2, \beta$ independent of h.

In practical implementations, $U_K^n = \|\boldsymbol{u}_h^n\|_{\mathbf{L}^2(K)}/|K|^{1/2}$, with $|K|$ denoting the surface ($d = 2$) or volume ($d = 3$) of element K, considering that in practice $\boldsymbol{u}_h$ solves the NSE. Also, in three dimensions, we define the element size $h_K = \sqrt[3]{|K|}$, for all $K \in \mathcal{T}_h$, for simplicity of implementation.

Note that when dealing with equal-order interpolations, a specific discrete inf–sup condition that is essential for the stability of the proposed method can be derived:

Lemma 3.2. *Assume that Hypothesis 3.1 holds. Then, for a uniformly regular family of triangulations $\{\mathcal{T}_h\}_{h>0}$, we have the following inf–sup condition:*

$$\forall q_h \in \mathbb{M}_h, \quad \|q_h\|_{L^2} \leqslant C \left(\sup_{\boldsymbol{v}_h \in \mathbf{X}_h} \frac{(\nabla \cdot \boldsymbol{v}_h, q_h)_\Omega}{\|D(\boldsymbol{v}_h)\|_{\mathbf{L}^2}} + \|\sigma_h^*(\nabla q_h)\|_{\tau_m} \right), \tag{3.10}$$

for some positive constant C independent of h, where τ_m denotes here the weighted L^2-norm with stabilization coefficient $\tau_{m,K}$.

The proof of this lemma can be derived from [8], where it is also shown that the discrete inf–sup condition (3.10) can be extended to a more complex condition that holds for a simply regular family of triangulations.

Remark 3.3. The presented high-order term-by-term stabilization procedure in space by using a Scott–Zhang-like interpolation operator has been extended to geophysical flows governed by the primitive equations of the ocean [43] and buoyant flows governed by the Boussinesq equations [44], giving high-order accuracy for relatively coarse grids. Also, it has been combined with a VMS–Smagorinsky term and wall laws for the accurate simulation of turbulent boundary layers in [45–47].

3.2. Time discretization: incremental pressure-correction algorithm with semi-implicit backward differentiation formulas

To state the proposed time discretization of the unsteady LPS model described by system (3.2), consider a positive integer number N and define $\Delta t = T/N$, $t_n = n\Delta t$, $n = 0, 1, \ldots, N$. We compute the approximations $\boldsymbol{u}_h^n$ and p_h^n to $\boldsymbol{u}^n = \boldsymbol{u}(\cdot, t_n)$ and $p^n = p(\cdot, t_n)$, respectively, by using an incremental pressure-correction scheme based on semi-implicit BDF, for which the nonlinear terms are extrapolated by means of Newton–Gregory backward polynomials [48]. Let us denote by $\widetilde{\boldsymbol{u}}_h^n$ an intermediate approximate velocity at time t_n. In order to abbreviate its discrete time derivative, we define the operator D_t by:

$$D_t^r \widetilde{\boldsymbol{u}}_{h^n} = \frac{\alpha_r \widetilde{\boldsymbol{u}}_{h^{n+1}} - \widetilde{\boldsymbol{u}}_{h,r^n}}{\Delta t}, \tag{3.11}$$

where for BDF schemes of orders $r = 1, 2$ we have:

$$\widetilde{\boldsymbol{u}}^n_{h,r} = \begin{cases} \widetilde{\boldsymbol{u}}^n_h & \text{if } n \geqslant 0, \ \text{for } r = 1 \ \text{(BDF1)} \\ 2\widetilde{\boldsymbol{u}}^n_h - \dfrac{1}{2}\widetilde{\boldsymbol{u}}^{n-1}_h & \text{if } n \geqslant 1, \ \text{for } r = 2 \ \text{(BDF2)} \end{cases} \tag{3.12}$$

and

$$\alpha_r = \begin{cases} 1, & \text{for } r = 1 \ \text{(BDF1)} \\ \dfrac{3}{2}, & \text{for } r = 2 \ \text{(BDF2).} \end{cases} \tag{3.13}$$

We consider the following extrapolations [48] of order $r = 1, 2$ for the intermediate convection velocity:

$$\widetilde{\boldsymbol{u}}^{n,*}_{h,r} = \begin{cases} \widetilde{\boldsymbol{u}}^n_h & \text{if } n \geqslant 0, \ \text{for } r = 1 \ \text{(BDF1)} \\ 2\widetilde{\boldsymbol{u}}^n_h - \widetilde{\boldsymbol{u}}^{n-1}_h & \text{if } n \geqslant 1, \ \text{for } r = 2 \ \text{(BDF2)} \end{cases} \tag{3.14}$$

and the pressure:

$$p^{n,*}_{h,r} = \begin{cases} p^n_h & \text{if } n \geqslant 0, \ \text{for } r = 1 \ \text{(BDF1)} \\ \dfrac{1}{3}(7p^n_h - 5p^{n-1}_h + p^{n-2}_h) & \text{if } n \geqslant 1, \ \text{for } r = 2 \ \text{(BDF2)} \end{cases} \tag{3.15}$$

using in the last case the convention $p^{n-1}_h = p^{n-2}_h$ for $n = 1$. In this way, after applying a standard incremental pressure-correction approach (cf. [42,49] for derivation in the case of equal-order and inf–sup stable FE, respectively) to system (3.2), the fully discrete semi-implicit formulation consists in solving, for $n = 0, \dots, N - 1$, the two-step algorithm:

find $\widetilde{\boldsymbol{u}}^{n+1}_h \in \mathbf{X}_h$ *such that:*

$$\begin{cases} \left(D_t^r \widetilde{\boldsymbol{u}}^n_h, \boldsymbol{v}_h\right)_\Omega + b(\widetilde{\boldsymbol{u}}^{n,*}_{h,r}, \widetilde{\boldsymbol{u}}^{n+1}_h, \boldsymbol{v}_h) + a(\widetilde{\boldsymbol{u}}^{n+1}_h, \boldsymbol{v}_h) + s_{\text{conv}}(\widetilde{\boldsymbol{u}}^{n,*}_{h,r}, \widetilde{\boldsymbol{u}}^{n+1}_h, \boldsymbol{v}_h) \\ + s_{\text{div}}(\widetilde{\boldsymbol{u}}^{n+1}_h, \boldsymbol{v}_h) = \langle \overline{\boldsymbol{f}}^{n+1}, \boldsymbol{v}_h \rangle + \left(p^{n,*}_{h,r}, \nabla \cdot \boldsymbol{v}_h\right)_\Omega, \end{cases} \tag{3.16}$$

for any $\boldsymbol{v}_h \in \mathbf{X}_h$, where $\overline{\boldsymbol{f}}^{n+1}$ is the average value of $\boldsymbol{f}$ in $[t_n, t_{n+1}]$, and:

find $p^{n+1}_h \in \mathbb{M}_h$ *such that:*

$$\begin{cases} \left(\nabla(p^{n+1}_h - p^n_h), \nabla q_h\right)_\Omega + s_{\text{pres}}(p^{n+1}_h, q_h) = -\dfrac{\alpha_r}{\Delta t}\left(\nabla \cdot \widetilde{\boldsymbol{u}}^{n+1}_h, q_h\right)_\Omega \\ \left(\boldsymbol{n} \cdot \nabla(p^{n+1}_h - p^n_h)\right)_{|\Gamma} = 0, \end{cases} \tag{3.17}$$

for any $q_h \in \mathbb{M}_h$, where $\boldsymbol{n}$ is the outer normal to Γ.

The final velocity can then be recovered according to:

$$\boldsymbol{u}^{n+1}_h = \widetilde{\boldsymbol{u}}^{n+1}_h - \frac{\Delta t}{\alpha_r} \nabla(p^{n+1}_h - p^n_h). \tag{3.18}$$

This is the temporal approach used in practical implementations. In particular, for the first time step ($n = 0$) we use a BDF1 scheme ($r = 1$) to initialize the algorithm with $\widetilde{\boldsymbol{u}}^0_h = \boldsymbol{u}^0_h$ and p^0_h some stable approximations to $\boldsymbol{u}^0$ and p^0, respectively. Note that this scheme coincides with a Backward Euler method. Then, a BDF2 scheme ($r = 2$) is applied for $n \geqslant 1$.

In order to achieve a global (space–time) second-order accuracy for velocities, $\mathbb{P}_2$ FE are used to approximate them in space. Thus, in the expressions (3.7) and (3.8) of the stabilization coefficients, $l = 2$. Also, $U^n_K = \|\widetilde{\boldsymbol{u}}^{n,*}_{h,r}\|_{\mathbf{L}^2(K)}/|K|^{1/2}$. Note that the term s_{pres} in Eq. (3.17) is neglected when dealing with inf–sup stable interpolations. This obviously leads to cheaper (amortized setup) Poisson solves for the pressure equation, maintaining a reasonable balance between accuracy and computational complexity. This is why, in the practical computations of this article, we choose mixed FE for which the pressure is approximated with $\mathbb{P}_1$ FE (i.e., Taylor–Hood FE pairs are considered).

Remark 3.4. The semi-implicit discretization in time segregating velocity and pressure through a standard incremental time-splitting helps to construct an efficient linear solver to the NSE system for the LES of turbulent flows. In the first step Eq. (3.16), a convection-dominated convection–diffusion–reaction problem for the intermediate velocity must be solved. The second step Eq. (3.17) consists of a (stabilized) pressure-Poisson problem. Both steps are solved by using a domain decomposition method preconditioner with the GMRES iterative method applied to the associated system in the parallel framework proposed in Section 4, and a convincing strong scaling analysis of the used algorithm is showcased in Section 5.2.1.

4. Parallel strategy for the fully discrete problem: interface with HPDDM

The fully discrete problem described in Section 3 gives rise to two large linear systems of the form $A^v u = f_v$ for subproblem (3.16) and $A^p q = f_p$ for subproblem (3.17), respectively. Solving such large linear systems could become extremely expensive from the computational point of view, that is why we adopt in the numerical implementation a highly parallel strategy based on domain decomposition methods.

To describe it, we first divide the mesh $\{\mathcal{T}_h\}_{h>0}$ in N non-overlapping meshes $\{\mathcal{T}_i\}_{1\leqslant i\leqslant N}$ using standard graph partitioners, e.g., ParMETIS [50]. If δ is a positive integer, the overlapping decomposition $\left\{\mathcal{T}_i^\delta\right\}_{1\leqslant i\leqslant N}$ is defined recursively as follows: $\mathcal{T}_i^\delta$ is obtained by including all elements of $\mathcal{T}_i^{\delta-1}$ plus all adjacent elements of $\mathcal{T}_i^{\delta-1}$. For $\delta = 0$, $\mathcal{T}_i^\delta = \mathcal{T}_i$. Let $\left\{\mathbf{X}_i^\delta\right\}_{1\leqslant i\leqslant N}$ (resp. $\left\{\mathbb{M}_i^\delta\right\}_{1\leqslant i\leqslant N}$) be the local velocity (resp. pressure) FE spaces defined on $\left\{\mathcal{T}_i^\delta\right\}_{1\leqslant i\leqslant N}$. Now, consider the restrictions $\left\{R_i^v\right\}_{1\leqslant i\leqslant N}$ (resp. $\left\{R_i^p\right\}_{1\leqslant i\leqslant N}$) from $\mathbf{X}_h$ (resp. $\mathbb{M}_h$) to $\left\{\mathbf{X}_i^\delta\right\}_{1\leqslant i\leqslant N}$ (resp. $\left\{\mathbb{M}_i^\delta\right\}_{1\leqslant i\leqslant N}$), and two local partitions of unity $\left\{D_i^v\right\}_{1\leqslant i\leqslant N}$ and $\left\{D_i^p\right\}_{1\leqslant i\leqslant N}$ such that:

$$\sum_{j=1}^{N} R_j^{v^{\mathrm{T}}} D_j^v R_j^v = Id_{n_v\times n_v} \qquad \sum_{j=1}^{N} R_j^{p^{\mathrm{T}}} D_j^p R_j^p = Id_{n_p\times n_p},$$

where Id denotes the identity matrix and n_v (resp. n_p) is the global number of unknowns in the velocity (resp. pressure) space. Algebraically speaking, if n_v is the global number of velocity unknowns and $\{n_{v_i}\}_{1\leqslant i\leqslant N}$ are the numbers of degrees of freedom in each local velocity FE space, then R_i^v is a Boolean matrix of size $n_{v_i} \times n$, and D_i^v is a diagonal matrix of size $n_{v_i} \times n_{v_i}$, for all $1 \leqslant i \leqslant N$. Of course, this also holds for the operators related to the pressure spaces.

Using the partition of unity, one can define the following one-level preconditioner as an extension of the Restricted Additive Schwarz (RAS) method proposed by Cai and Sarkis [51]:

$$\mathcal{M}_{\mathrm{ORAS}}^{v^{-1}} = \sum_{i=1}^{N} R_i^{v^T} D_i^v B_i^{v^{-1}} R_i^v, \tag{4.1}$$

where the $\left\{B_i^v\right\}_{1\leqslant i\leqslant N}$ are local operators that resemble the submatrices $\left\{R_i^v A^v R_i^{v^T}\right\}_{1\leqslant i\leqslant N}$, but with more efficient transmission conditions between subdomains, e.g., see [52]. In practice, we use simple Robin boundary conditions. This produces a rather efficient preconditioner for the successive linear systems related to the velocity unknowns even at large scale, when considering thousands of subdomains and millions of degrees of freedom, cf. Section 5.2.1. Note that the linear systems related to the velocity Eq. (3.16) are non-symmetric. Their sparsity pattern is however the same throughout the simulation as soon as the time discretization is properly initialized, as proposed in Section 3.2. Since we use in all simulations direct solvers to compute the action of the inverse of the local $\left\{B_i^v\right\}_{1\leqslant i\leqslant N}$, needed in Eq. (4.1), we can perform a single symbolic factorization after startup, followed by as many numeric factorizations as needed.

The same formalism cannot exactly be used as it is for solving the linear systems related to the pressure Eq. (3.17). First, note that for these systems only the right-hand sides are changing when using inf–sup stable FE, as it is the case in the present computations, for which the pressure stabilization term s_{pres} in Eq. (3.17) is neglected. We thus need to assemble a single matrix and setup a preconditioner once. These costs will then be amortized over the many successive time steps. Even with more complex transmission conditions between subdomains, a so-called "one-level" method such as $\mathcal{M}_{\mathrm{ORAS}}^{v^{-1}}$ defined in Eq. (4.1) would not scale properly in terms of number of iterations for solving a pressure-Poisson problem. In this case, we thus chose to use a more sophisticated multilevel domain decomposition method using the GenEO-2 approach [16]. This preconditioner, $\mathcal{M}_{\mathrm{GenEO\text{-}2}}^{p^{-1}}$, uses a spectral coarse grid to better couple all subdomains. Thus, as clearly shown in Section 5.2.1, the number of iterations for solving the linear systems related to the pressure unknowns remains really low even when considering large number of subdomains.

The applied domain decomposition methods are implemented in the library HPDDM [15,53] (more details about the implementation of the cited spectral preconditioners are given in [53]). HPDDM can be interfaced with various programming languages and open-source FE libraries such as FreeFem++ [17], the one used in the present simulations.

5. Numerical studies

In this section, we present two numerical experiments respectively based on a two- and a three-dimensional lid-driven cavity. The former has the aim of validating the proposed method, the latter of evaluating its numerical performances at laminar, transient, and turbulent regimes, also on massive parallel settings.

5.1. 2D lid-driven cavity flow

The two-dimensional lid-driven cavity test is one of the most popular validation problems for fluid flow simulations. In this test, the fluid is contained in a unitary squared domain and it has Dirichlet boundary conditions on all sides: three stationary sides and one moving side at the top, characterized by a unitary tangent velocity. The right-hand side of the momentum equation vanishes in Ω, i.e., $\mathbf{f} = \mathbf{0}$.

Three different flow conditions have been simulated, respectively for Reynolds numbers $Re = 1{,}000, 5{,}000, 10{,}000$ ($Re = 1/\nu$). In all cases, an impulsive start is performed, i.e., the initial condition is a zero velocity field, and the time step is $\Delta t = 0.01$. All computations are carried out with inf–sup stable FE of Taylor–Hood-type, i.e., $\mathbb{P}_2$-$\mathbb{P}_1$ FE are used to approximate the velocity–pressure pair on grids refined towards the walls in both spatial directions using the hyperbolic tangent function:

$$f(x) = 0.5\left\{1 + \frac{\tanh[2(2x-1)]}{\tanh(2)}\right\}. \tag{5.1}$$

The partition of the cavity is 32^2, 50^2, 64^2 respectively for the Reynolds numbers $Re = 1{,}000, 5{,}000, 10{,}000$. Numerically, a steady solution seems to exist up to $Re = 7{,}500$ approximately. Bifurcations to periodic unsteady solutions, obtained by DNS simulations, are reported by many authors, e.g., [54,55]. So, for $Re = 1{,}000$ and $5{,}000$, we assume that a stationary solution is reached if the velocity deviation between two consecutive time steps is lower than a chosen tolerance, fixed as:

$$\|\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n\|_{\mathbf{L}^2} \leqslant 10^{-6}.$$

For $Re = 10{,}000$, we just let run the simulation till a final time $T = 1{,}000$. To compare our results, we consider those of Erturk et al. [56] obtained using the streamfunction–vorticity formulation on a very fine uniform grid mesh of 601^2.

The most interesting feature is the genesis of a varying number of vortices inside the cavity. Fig. 1 shows the streamlines on a colored velocity distribution for all the Reynolds numbers considered, $Re = 1{,}000, 5{,}000, 10{,}000$, at the final computation time. We observe that secondary and tertiary vortices are well-resolved. In Fig. 2, we show cross-sections of the horizontal and vertical final velocities, respectively at $y = 0.5$ and $x = 0.5$, for all the Reynolds numbers considered. Hereafter, in the legends of figures, we denote by BDF2 FS-LPS our proposed method. It can be observed that the results for the used coarse grids are in agreement with the ones obtained by Erturk et al. [56], also in the boundary layers, thus validating the method.

5.2. 3D lid-driven cavity flow

In this section, the three-dimensional lid-driven cavity test is performed to investigate the numerical performances of the proposed solver at laminar, transient, and turbulent regimes, also on massive parallel settings. Similarly to the two-dimensional problem, the three-dimensional case is characterized by a fluid flow in a cubic domain driven by a tangential unitary velocity along one of the six boundary surfaces. Homogeneous Dirichlet conditions are adopted on all the other boundaries.

The recirculating flow in a 3D lid-driven cavity presents the occurrence of some considerable 3D features, even at relatively low Reynolds numbers. One of the most remarkable is the formation of Taylor–Görtler-like (TGL) vortices at the corners of the bottom of the cavity. Small counter-rotating vortices are formed as a result of the curvature of the streamlines due to the main vortex in the middle of the cavity [57], similarly to the 2D case, but with a genuine 3D structure. Zang et al. [58] report the numerical simulation of the 3D cavity flow by a LES model using the Finite

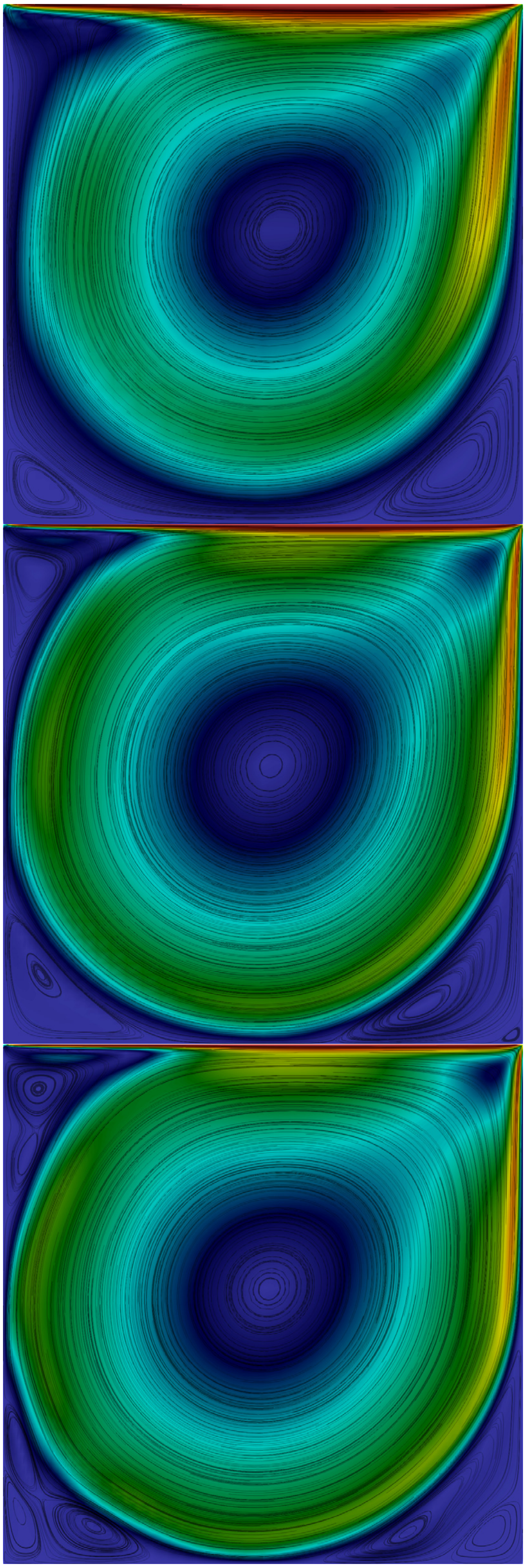
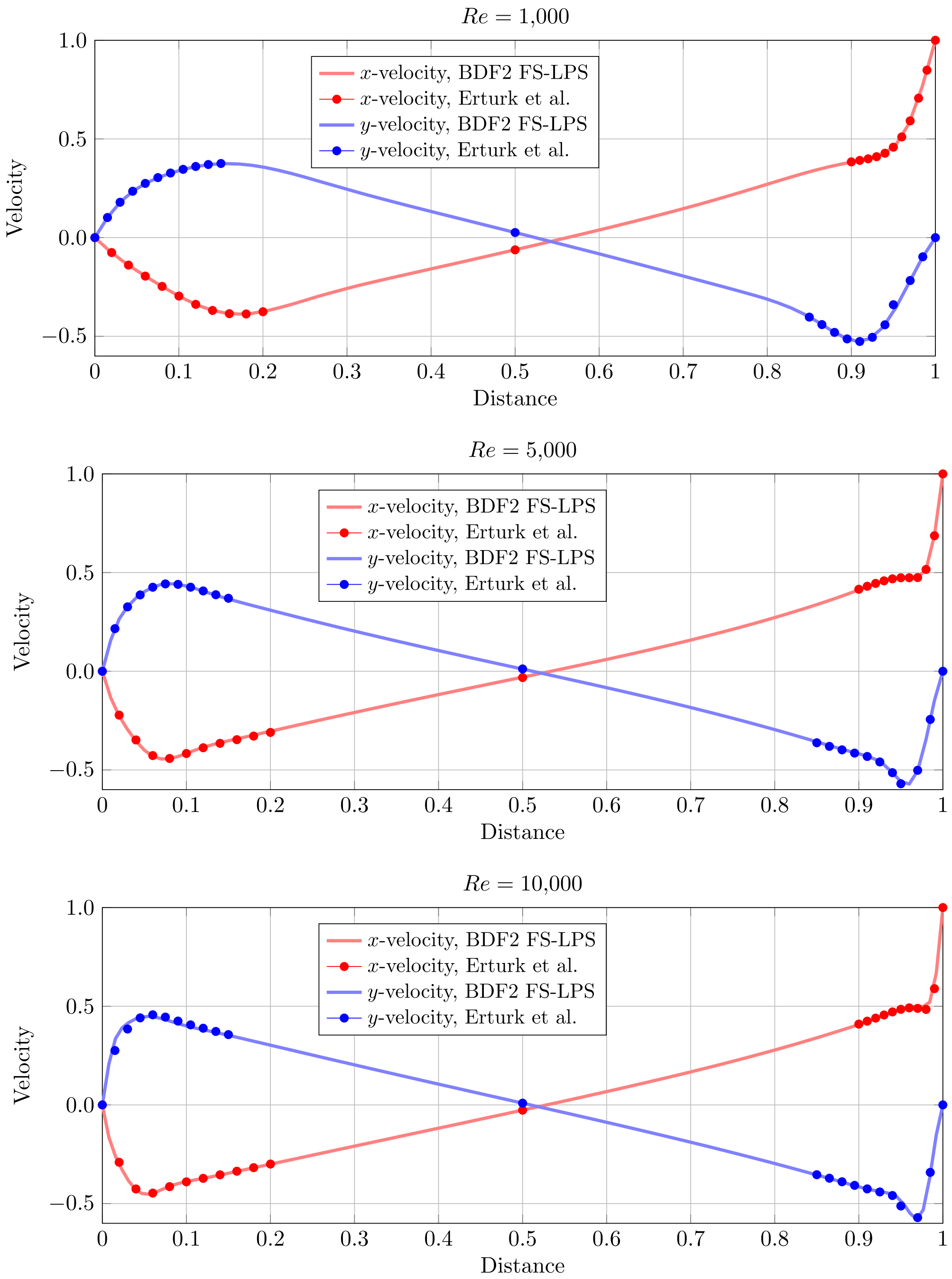


Fig. 1. Streamlines on colored velocity distribution at $Re = 1{,}000$, 5,000, 10,000 (from top to bottom). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



$Re = 1{,}000$

$Re = 5{,}000$

$Re = 10{,}000$

Fig. 2. Comparison of velocity profiles along vertical and horizontal lines through geometric center with respect to data of Erturk et al. [56].

Volume Method (FVM), using the dynamic procedure of Germano et al. [59]. Based on experimental experiences performed in Prasad and Koseff [60], they describe the flow at Reynolds number 3,200 to be essentially laminar, although an inherent unsteadiness may occur. For Reynolds number 7,500, a transitional stage is reached, since the flow becomes unstable near the downstream eddies at Reynolds numbers higher than about 6,000. With even higher Reynolds numbers at about 10,000, the flow becomes fully turbulent. Thus, laminar, transient, and turbulent regimes are covered by the choice of these three cases.

The primary goal of the simulation of the 3D cavity flow is to obtain a bounded kinetic energy as time increases, during the complete simulation time needed to reach a stable equilibrium [61]. This may look as a simple requirement, but some turbulence models violate it. Indeed, Iliescu et al. [62] reported the numerical results obtained with three subgrid scale models for $Re = 10{,}000$: the Smagorinsky model [63], a traditional Taylor LES model of Clark et al., [64], and two variants of a new rational LES model developed in Galdi and Layton [65]. It was shown that the Taylor LES model produces an energy blow-up in finite time. The two rational LES models did not cause an energy blow-up, but exhibited important oscillations. The standard Smagorinsky model (with the Smagorinsky constant $C_S = 0.1$) turned out to be notably more diffusive, as expected.

Following the works of Gravemeier et al. [66] and Zang et al. [58], we simulate the 3D cavity flow at Reynolds numbers $Re = 3{,}200$, 7,500, 10,000. In [66], Gravemeier et al. analyze the performances of a VMS model based on the Residual Free Bubbles (RFB) method and the use of several nested meshes. This is called the three-level FE method (VMS-3L), as it includes three grid levels, and it takes into account the effect of small unresolved scales onto small resolved scales by a subgrid eddy viscosity approach in Smagorinsky's form.

A first difficulty we face in the numerical simulations is to obtain a high-order accuracy with a relatively coarse basic discretization (i.e., low computational cost), for all flow situations. The computational grid consists of a 32^3 partition of the unit cube, uniform in the y-direction, and refined towards the walls in both the x- and z-directions using the hyperbolic tangent function (5.1), in order to handle large velocity gradients. This already provides a large improvement in the accuracy of the numerical results. On this mesh, we consider inf–sup stable three-dimensional FE. In particular, Taylor–Hood $\mathbb{P}_2$-$\mathbb{P}_1$ FE are used to approximate the velocity–pressure pair (no need to consider pressure stabilization). In any case, we are considering a number of degrees of freedom comparable to the one of the VMS-3L numerical simulation of Gravemeier et al. [66]. For all Reynolds numbers considered, an impulsive start is performed, i.e., the initial condition is a zero velocity field, and $\Delta t = 0.1$, as in [62,66]. A characteristic time scale T_{cav} is defined in Zang et al. [58] to be the estimated time for a fluid particle at the edge of the top boundary layer to turn and (approximately) reach its starting position in the cavity. This time scale is roughly estimated to be about 10 time units for the current computations. Initially, the simulation is run for five time scales T_{cav}, i.e., 50 time units or 500 time steps. Within this time period, the flow is expected to develop to full extent [66], including a subsequent relaxation time. Afterwards, statistics (quasi-steady results) are collected for another five time scales T_{cav}.

When available, results are graphically compared to the experimental data of Prasad and Koseff [60], and numerical results of Gravemeier et al. [66]. The experimental data for the flow at Reynolds number $Re = 7{,}500$ have only been evaluated for half of the cavity.

The temporal evolution of the total kinetic energy subject to:

$$E_{kin}(\boldsymbol{u}_h^n) = \frac{1}{2}\int_{\Omega} \boldsymbol{u}_h^n \cdot \boldsymbol{u}_h^n \, d\boldsymbol{x},$$

for the three cases $Re = 3{,}200$, 7,500, and 10,000 is displayed in Fig. 3. The flows become roughly stationary at $t \simeq 5\,T_{cav}$ (i.e., at about 50 time units), as expected. These results are almost comparable with the ones obtained by the VMS-3L method in Gravemeier et al. [66].

The mean velocities $\langle u_1 \rangle$ and $\langle u_3 \rangle$ are computed as a discrete time average according to:

$$\langle u_i \rangle(\boldsymbol{x}) = \frac{1}{N/2}\sum_{n=N/2}^{N-1} u_i(\boldsymbol{x}, t_n), \qquad i = 1, 3, \qquad N = \#\,\text{time steps} = 1{,}000.$$

Fig. 4 shows the mean velocity $\langle u_1 \rangle$ on the centerline $z = 0.5$ of the longitudinal mid-plane $y = 0.5$, for the various Reynolds numbers under consideration. The proposed method is in agreement with the experimental data of Prasad and Koseff [60], even with the coarse basic discretization at hand, and performs better than the VMS-3L method [66]. A similar accuracy is found for the mean velocity $\langle u_3 \rangle$ on the centerline $x = 0.5$ of the longitudinal mid-plane $y = 0.5$, see Fig. 5.

Also, higher-order moments $\langle \tilde{\boldsymbol{u}}^n \rangle$, with $n > 1$ and $\tilde{\boldsymbol{u}}$ denoting the fluctuating part of $\boldsymbol{u}$, are achieved by collecting values in the sense of a discrete time average, which is an appropriate procedure for stationary turbulence. In particular, we have considered the variance ($n = 2$) for the first and third component of the velocity, that reads $\langle \tilde{u}_i^2 \rangle = \langle u_i^2 \rangle - \langle u_i \rangle^2$, with the standard deviation (root mean square, RMS) defined as $\sqrt{\langle \tilde{u}_i^2 \rangle}$ $(i = 1, 3)$. Also, the off-diagonal component $\langle \tilde{u}_1 \tilde{u}_3 \rangle = \langle u_1 u_3 \rangle - \langle u_1 \rangle \langle u_3 \rangle$ of the Reynolds stress tensor is depicted, and finally the turbulent kinetic energy (TKE) defined as $\frac{1}{2}\sum_{i=1}^{3} \langle \tilde{u}_i^2 \rangle$ is reported, together with the corresponding energy spectrum. As in Prasad and Koseff [60],

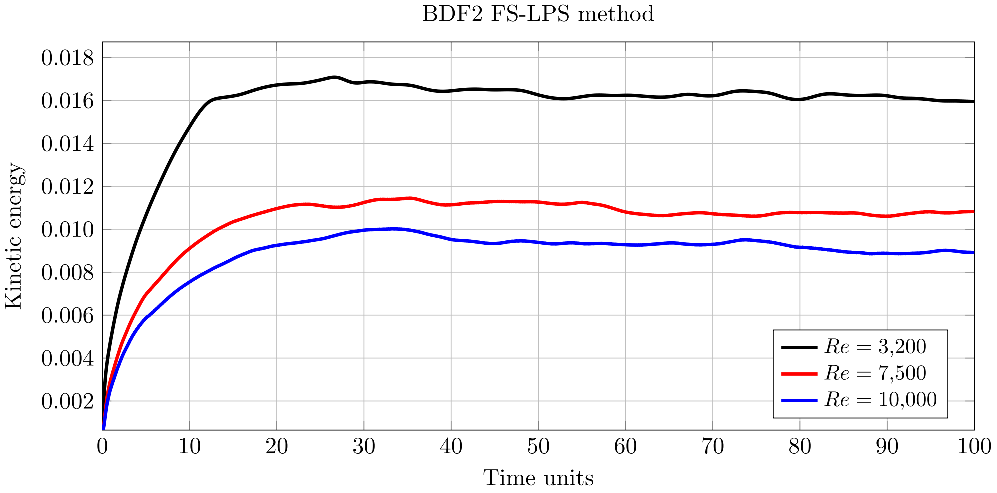


Fig. 3. Temporal evolution of the total kinetic energy for Re = 3,200, 7,500, 10,000 (from top to bottom).

the RMS values and the off-diagonal Reynolds stress component are multiplied by the amplification factors 10 and 500, respectively, in order to ensure a reasonable visual impression of these values within the respective graphs, and analogously the TKE is multiplied by the amplification factor 100. With respect to the experimental data, larger errors appear for the RMS values and the crossed component of the Reynolds stress tensor. These deviations are shown for Re = 10,000 in Figs. 6 and 7. Mispredictions of various peaks of these curves may also be found in the numerical results of Zang et al. [58], achieved with a two-time finer discretization in every coordinate directions. This underlines the difficulty in predicting these sensitive measures. For completeness, we also report the TKE obtained with our scheme on the centerlines of the mid-plane $y = 0.5$ for Re = 10,000 in Fig. 8, although experimental data and numerical results from Gravemeier et al. [66] are not available for this metric. From the TKE, we derive the corresponding energy spectrum on the centerlines of the mid-plane $y = 0.5$ for Re = 10,000 in Fig. 9, where we denote by $E(k)$ the energy spectrum, and k is the wavenumber. The slope of the energy spectrum in the inertial subrange is in line with the one of the theoretical Kolmogorov curve $k^{-5/3}$.

Qualitatively, we have observed that the flow exhibits effectively the formation of three-dimensional TGL corner vortices at the cavity end walls, that interact with the primary circulation vortex, thus influencing the distribution of momentum within the entire cavity, see Fig. 10. In the case $Re = 3200$, in accordance to Prasad and Koseff [60], it is possible to discern these vortices as organized structures, while for higher Re, increasing turbulent effects cause the breakdown of these organized structures, resulting in a “weaker” flow when compared with the pure two-dimensional flow. This suggests that the high-frequency turbulent fluctuations become dominant, and they partially destroy the integrity (or coherence) of the TGL vortices.

5.2.1. *Parallel performances of the solver*

The goal of this section is to assess the parallel efficiency of the proposed method. In particular, we are interested in the strong scalability of the described algorithm. In all that follows, we start by generating a global mesh using 100 grid points in each direction, distributed in the x- and z-directions using the hyperbolic tangent function (5.1). The mesh is then decomposed using ParMETIS [50]. The operations related to the discretization of continuous operators are performed using FreeFem++ [17]. The linear solvers and preconditioners are implemented in HPDDM [53]. Internally, the underlying linear solver for all local subdomain solves is Intel MKL PARDISO [67,68]. The stabilization terms from Eq. (3.6) have to be evaluated at each time step but involve purely local computations. In our experiments, the velocity field was initially set to zero. Thus, it takes one time step for the sparsity pattern of S to remain the same. We exploit this property by computing the matrix–matrix–matrix product first by using the domain-specific language of FreeFem++, and then by using PETSc [69] and its MatPtAPNumeric[1] routine (where the symbolic product may be

[1] www.mcs.anl.gov/petsc/petsc-current/docs/manualpages/Mat/MatPtAPNumeric.html.

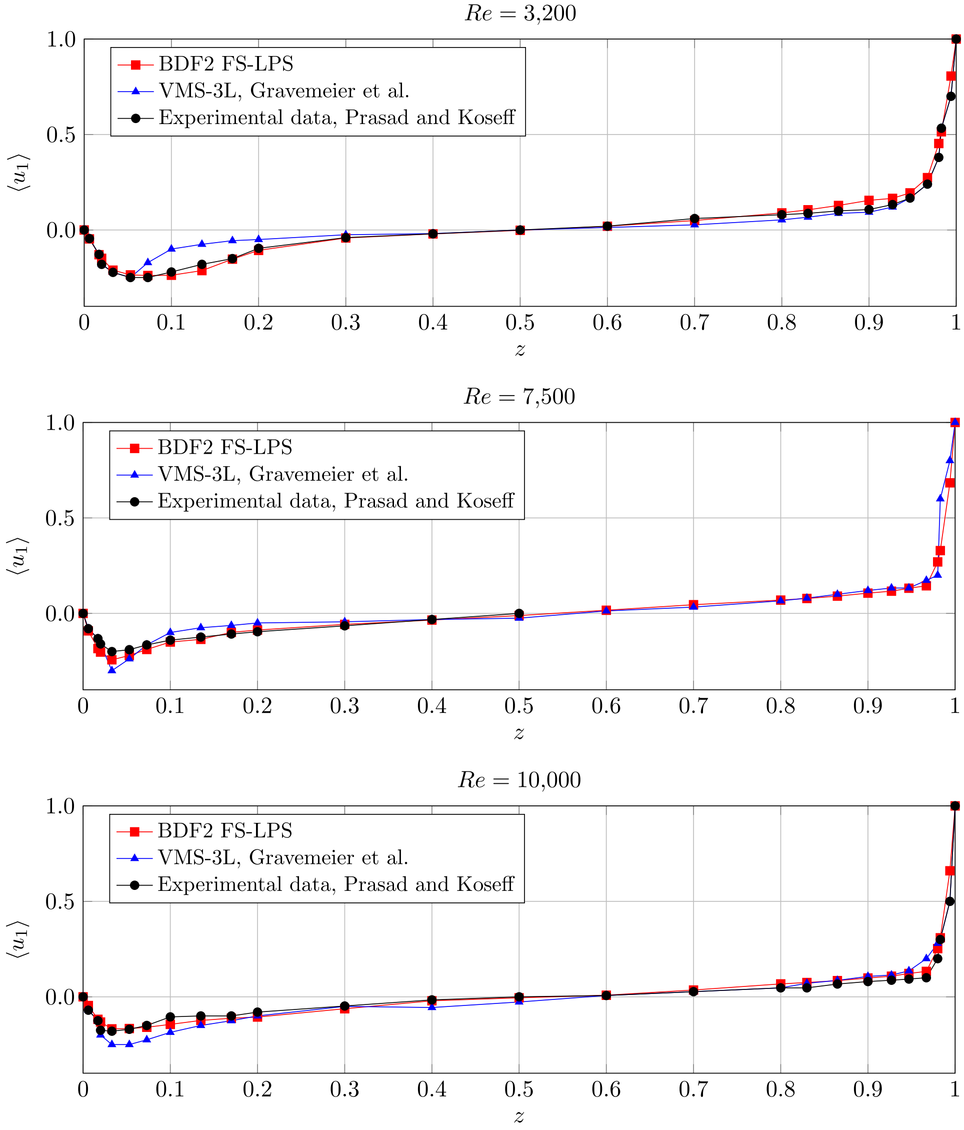


Fig. 4. $\langle u_1 \rangle$ on the vertical centerline ($z = 0.5$) of the mid-plane $y = 0.5$ for Re = 3,200, 7,500, 10,000.

bypassed). The matrix–matrix product $P \cdot D$, with D expressed as a sum of two or three matrices, is computed using a dedicated routine. Results were obtained on Curie, a system composed of 5,040 nodes with two eight-core Intel Sandy Bridge clocked at 2.7 GHz. The interconnect is an InfiniBand QDR full fat tree and the MPI implementation exploited was IntelMPI version 2017.0.2.174. All binaries and shared libraries were compiled with Intel compilers and Math Kernel Library support (for dense linear algebra computations). We used from 512 up to 16,384 MPI processes with a single OpenMP thread per process (flat MPI parallelism). Preconditioners are thus defined with as many subdomains as the number of MPI processes.

We used a test case in the turbulent regime with Re = 15,000 and $\Delta t = 0.05$. The successive discretizations of Eq. (3.16) yield linear systems $A^v u = f_v$ with 24.7×10^6 unknowns, while the single discretization of Eq. (3.17) yields a linear system $A^p q = f_p$ with 1.1×10^6 unknowns. Both systems are left-preconditioned by either $\mathcal{M}_{\text{ORAS}}^{v^{-1}}$ or $\mathcal{M}_{\text{GenEO-2}}^{p^{-1}}$ and the GMRES method is stopped when the relative preconditioned residual is lower than 10^{-8} for the velocity unknowns and 10^{-6} for the pressure unknowns. Since we have observed in the two previous sections that the

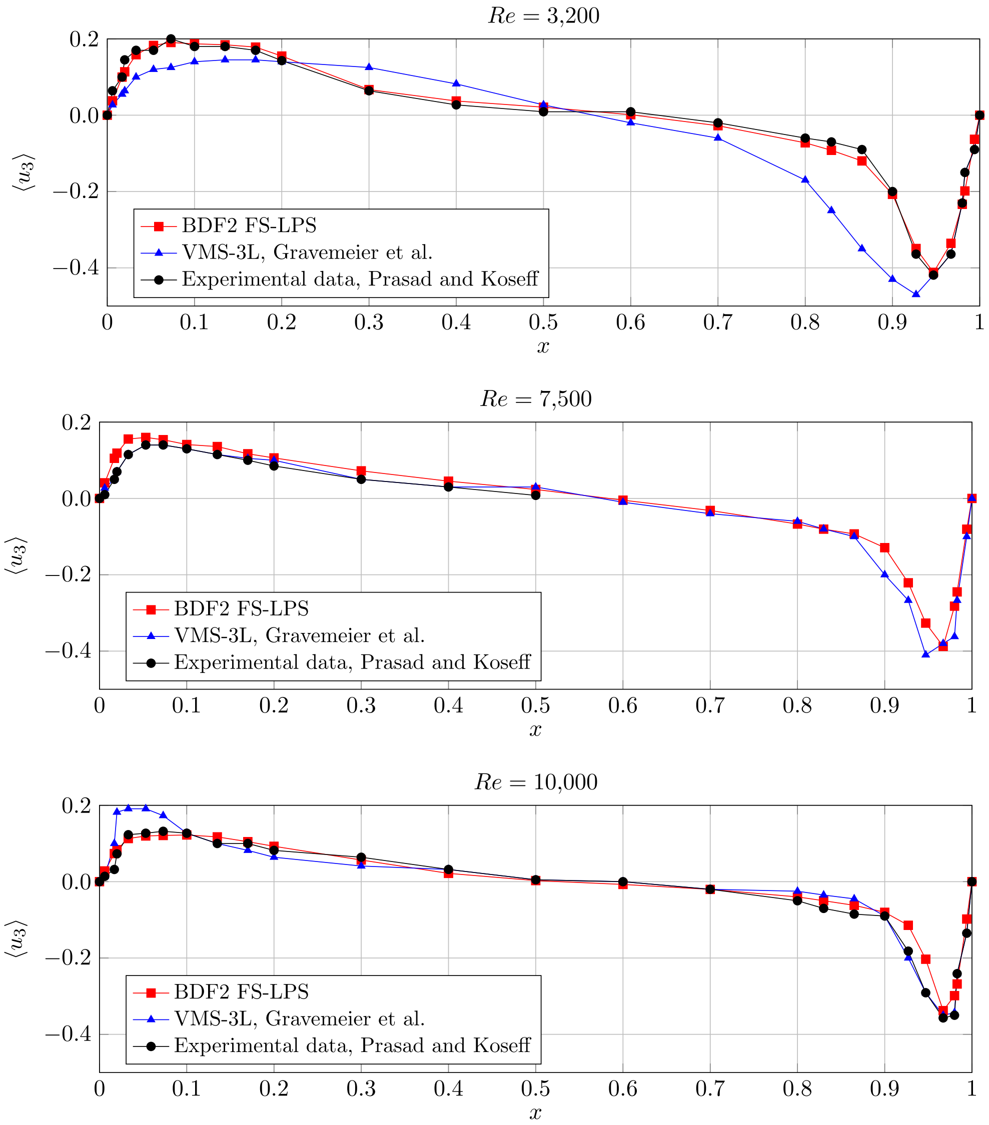


Fig. 5. $\langle u_3 \rangle$ on the horizontal centerline ($x = 0.5$) of the mid-plane $y = 0.5$ for Re = 3,200, 7,500, 10,000.

proposed solver is stable throughout the time steps, we will consider in the present scalability analysis only the 10th time step. This means that the startup phase is over. Since the single construction of $\mathcal{M}^{p^{-1}}_{\text{GenEO-2}}$ is amortized over time, it will not be included in the present analysis.

In order to give a complete overview of the performances for the proposed method, we first represent in Fig. 11 the total time to complete the 10th time step. Clearly, the implementation scales very well. This may be explained by multiple facts. As displayed in Fig. 12, the preconditioners are indeed both numerically extremely stable, with numbers of iterations remaining in the same low range. In Table 1, we report the time spent in all subroutines of the 10th time step. The first column (N) represents the number of subdomains (or MPI processes), to which most of subroutines scale almost linearly. The second column (stabilization) is the time needed by FreeFem++ and PETSc to assemble the stabilization terms. The third column (optimized operators) is the sum of the time spent assembling the velocity linear system and adding the optimized boundary conditions used for the preconditioner $\mathcal{M}^{v^{-1}}_{\text{ORAS}}$. The fourth column (BDF2 & aux. tasks) represents the amount of time spent incrementing the time step, mostly vector summations, as well as auxiliary tasks involving some communications, e.g., updating ghost elements. Since we are using exact LU

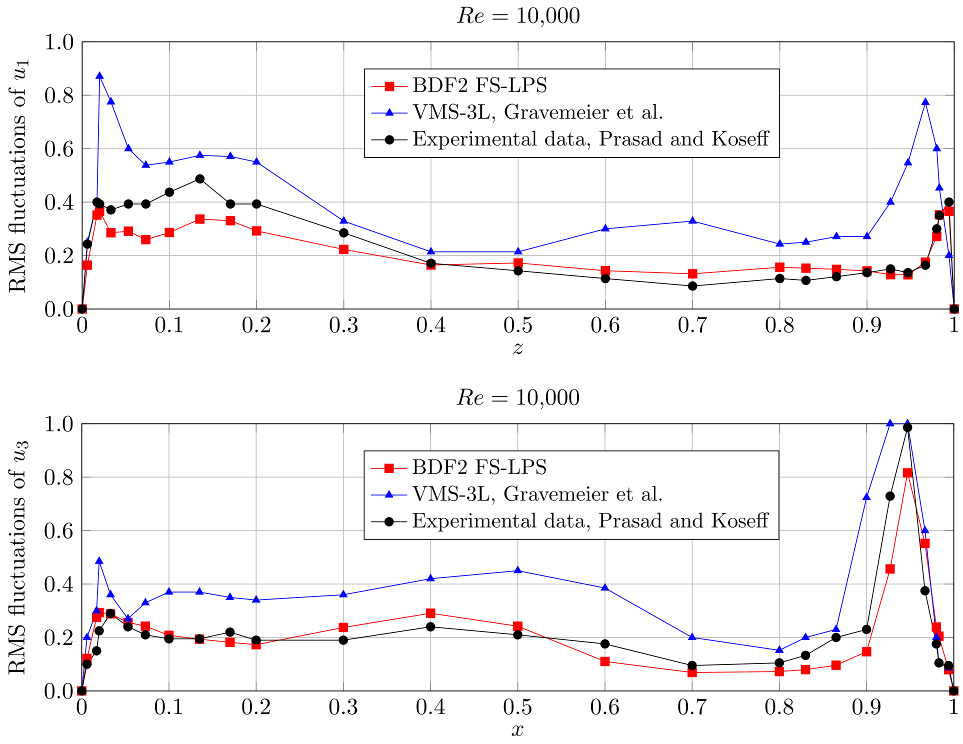


Fig. 6. $\sqrt{\langle \tilde{u}_1^2 \rangle}$ and $\sqrt{\langle \tilde{u}_3^2 \rangle}$ on the centerlines of the mid-plane $y = 0.5$ for $Re = 10{,}000$ (factor 10).

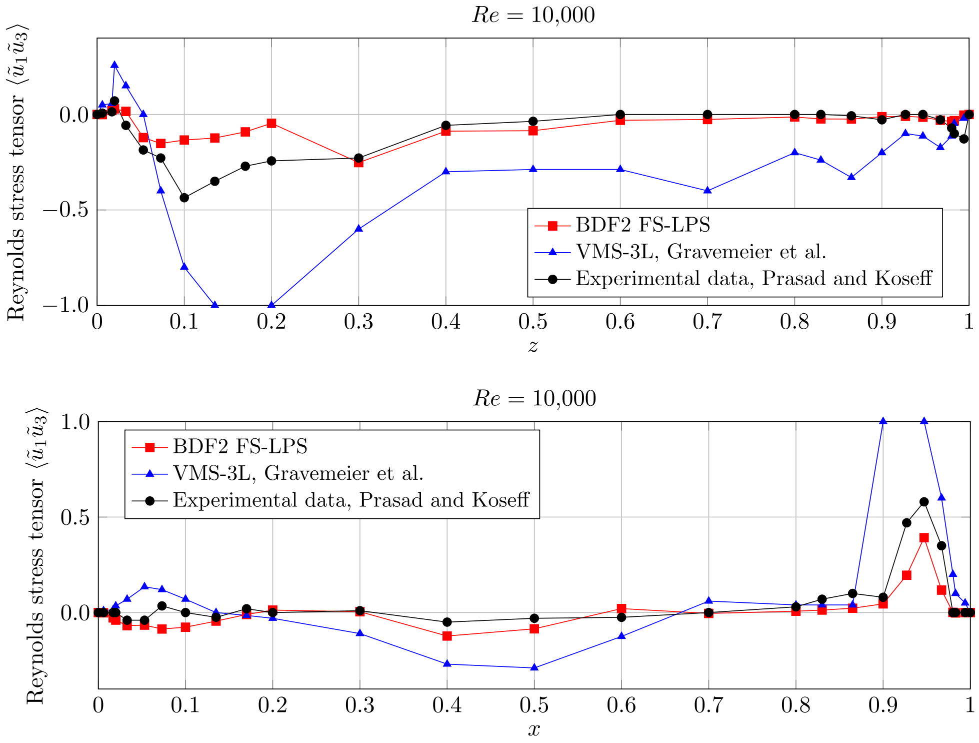


Fig. 7. $\langle \tilde{u}_1 \tilde{u}_3 \rangle$ on the centerlines of the mid-plane $y = 0.5$ for $Re = 10{,}000$ (factor 500).

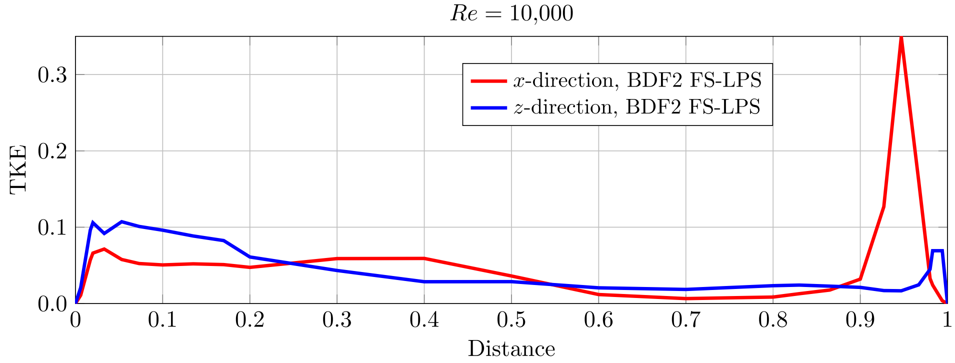


Fig. 8. TKE on the centerlines of the mid-plane $y = 0.5$ for $Re = 10{,}000$ (factor 100).

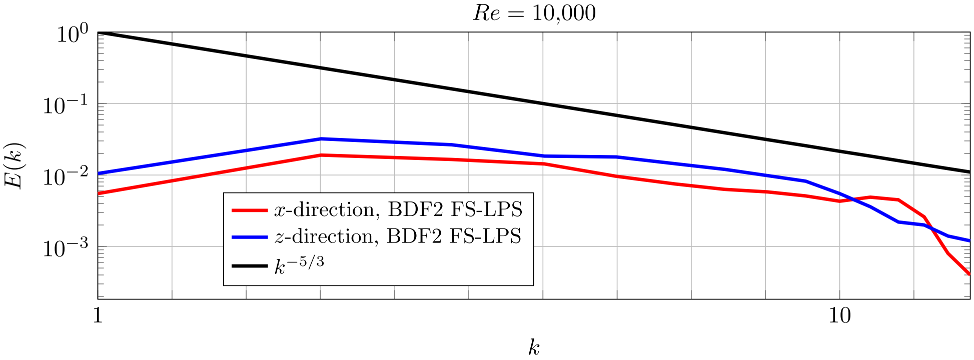


Fig. 9. Energy spectrum in logarithmic scale on the centerlines of the mid-plane $y = 0.5$ for $Re = 10{,}000$.

Table 1
Breakdown of the time spent in the various subroutines involved in the scalability analysis of Fig. 11 for a single time step. A^v and A^p are of order 24 millions and 1 million, respectively.

N	Stabi-lization	Optimized operators	BDF2 & aux. tasks	Setup $\mathcal{M}_{\text{ORAS}}^{v^{-1}}$	Solve $A^v u = f_v$	Solve $A^p q = f_p$	Total
512	16.8 s	39.8 s	2.0 s	140.8 s	7.8 s	0.1 s	207.4 s
1,024	7.5 s	22.8 s	1.2 s	40.4 s	6.0 s	0.2 s	78.0 s
2,048	4.6 s	11.6 s	0.7 s	13.3 s	2.7 s	0.2 s	33.1 s
4,096	2.1 s	7.0 s	0.4 s	4.4 s	1.5 s	0.3 s	15.9 s
8,192	1.5 s	4.4 s	0.3 s	2.4 s	0.9 s	0.7 s	10.2 s
16,384	0.9 s	2.8 s	0.2 s	1.1 s	0.4 s	0.8 s	6.1 s

decompositions for subdomain solves, we can see that the fifth column (setup $\mathcal{M}_{\text{ORAS}}^{v^{-1}}$) exhibits a super-linear speedup. It is also worthwhile to note that the solution phase for the pressure unknowns (seventh column) does not scale as the solution phase for the velocity unknowns (sixth column). Indeed, because of the use of a multilevel preconditioner, the volume of communication tends to outgrow the number of local computations, which decreases in a strong scaling experiment. There is no such phenomenon for the velocity unknowns since we are using a simpler one-level domain decomposition methods with much less communication.

In Fig. 13, we display the relative time (in percent) spent in each subroutines of Table 1. The first three colors represent tasks that are purely concurrent and that do not involve any kind of communication. The last three colors represent tasks with some communication between processes. For example, the GMRES method requires global reductions at each inner iteration for computing scalar products. Overall, we can see that even for 16,384 processes,

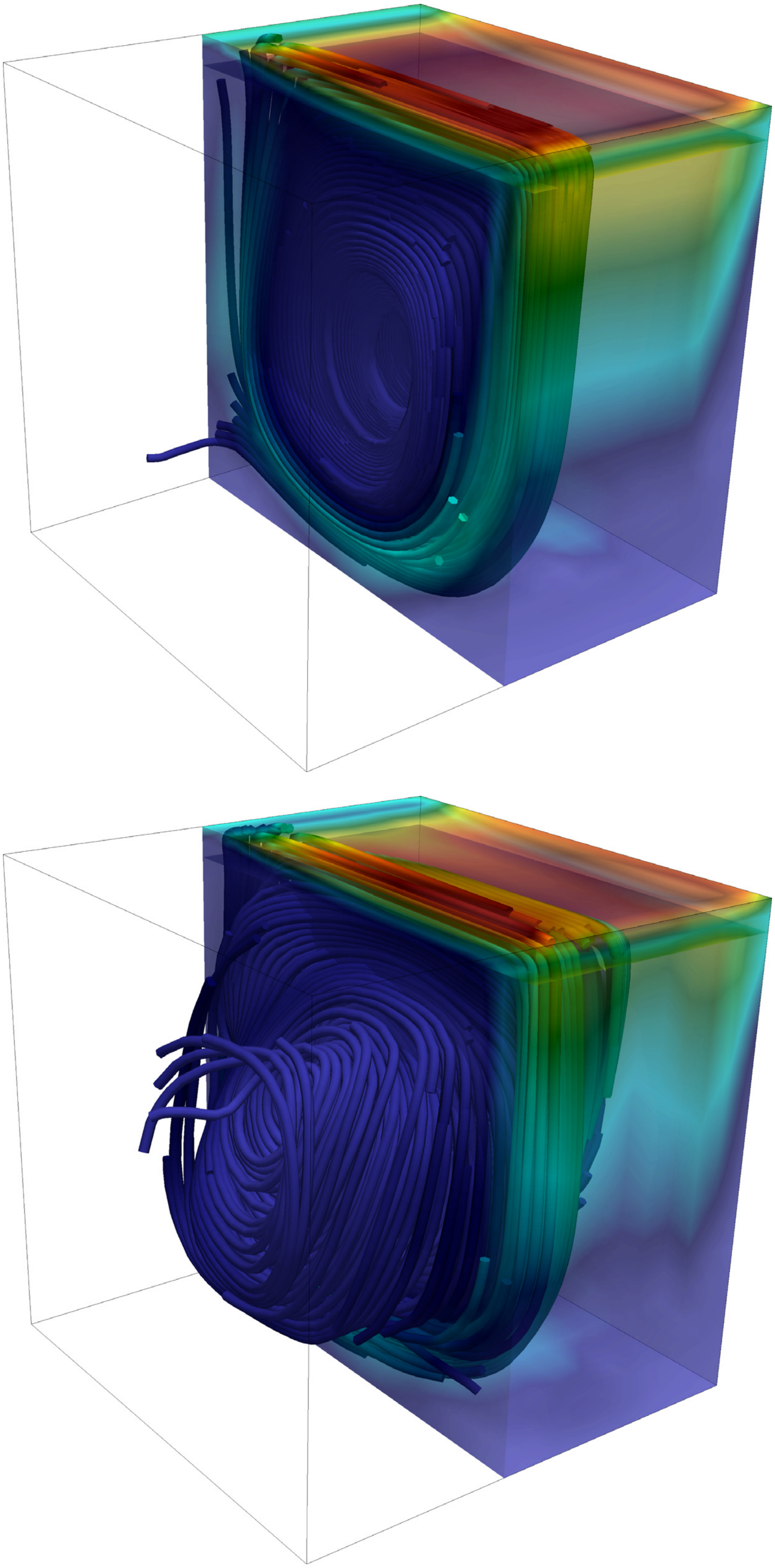


Fig. 10. Flow streamlines at $Re = 3{,}200$ (top) and $Re = 7{,}500$ (bottom); results for the proposed BDF2 FS-LPS method at final simulation time $T = 100$.

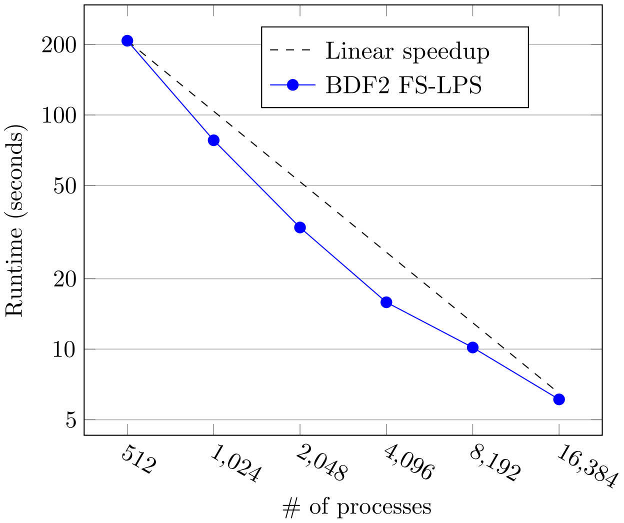


Fig. 11. Strong scaling analysis of the BDF2 FS-LPS implementation in 3D for a problem of 24 million velocity unknowns and 1 million pressure unknowns.

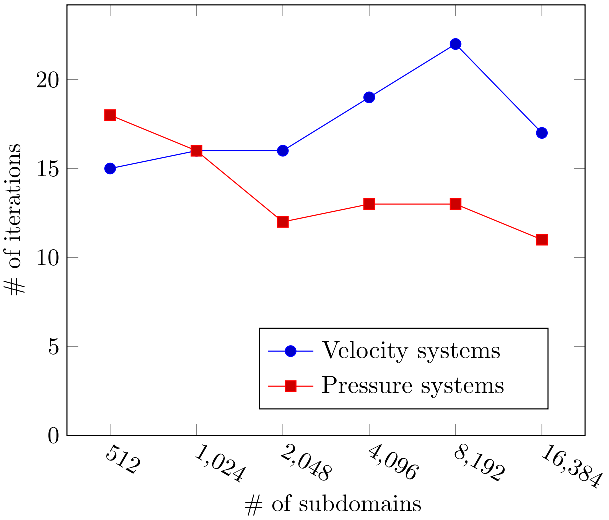


Fig. 12. Number of iterations for the GMRES method to reach convergence for a 3D problem of 24 million velocity unknowns and 1 million pressure unknowns.

most of the time (around 80% of the total) of a single time step is spent on local computations. This explains why the overall scalability of the proposed method is satisfactory on a wide range of process counts.

To sum up, the parallel performances of the proposed method are rather satisfactory, and seem to be in accordance with the current state-of-the-art, e.g., [18].

6. Summary and conclusions

In this work, we have proposed a FE LPS spatial approximation of the NSE, combined with an efficient velocity–pressure segregation, using semi-implicit BDF in time. In order to face the computational complexity of the problem, we have developed an ad-hoc parallel solver for the proposed method, based on HPDDM.

We have tested the proposed numerical scheme by solving the benchmark problem of the recirculating flow in a lid-driven cavity at high Reynolds numbers. This numerical study shows that the solver is able to reproduce first and

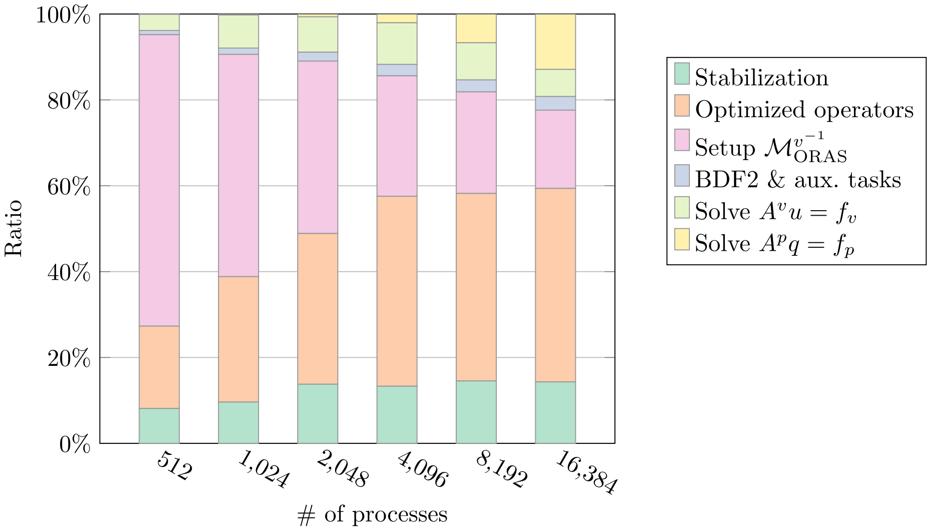


Fig. 13. Comparison of the time spent in the various subroutines involved in the scalability analysis of Fig. 11 for a single time step. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

second-order statistics up to a turbulent regime for relatively coarse meshes, with a similar (or even higher) accuracy than a more complex VMS–LES method [66]. We studied the practical performances of the solver implemented in a HPC framework, showing strong scalability results up to thousands of cores. This suggests that the present method is efficient, and also provide a reasonable compromise between accuracy and computational complexity, thus proposing it as a suitable and useful tool in the challenging simulation of turbulent flows.

Finally, an extension of the present approach is possible in the following directions:

- problems with outflow boundary conditions. In particular, we aim to compare in the framework proposed in this paper the rotational incremental pressure-correction scheme, cf. [70], to an alternative strategy proposed in [71], which has demonstrated to improve the accuracy for the standard incremental version while remaining compatible with the rotational one;
- enhancement of inf–sup stable FE to exactly divergence-preserving schemes, cf. [72]. This will eventually allow to remove grad–div stabilization;
- further improvement of the velocity subgrid model based on the local projection of the streamline derivative in the proposed parallel framework, maybe combined with a VMS-eddy viscosity term and wall laws [46] for a more accurate simulation of turbulent boundary layers in particular, and more complex turbulent flows in general;
- coupled flow models like non-isothermal incompressible flows. A step forward this direction has been recently done in [44], and the construction of an efficient solver for the method introduced in that paper is today in preparation, following the guidelines given in the present work.

Acknowledgments

The authors acknowledge Prof. F. Hecht and F. Nataf for the fruitful discussions, suggestions, and advice. S. Rubino gratefully acknowledge the financial support received from IdEx Bordeaux (Excellence Initiative of Université de Bordeaux) and FSMP (Fondation Sciences Mathématiques de Paris) during his postdoctoral research involved in this article. R. Haferssas partially carried out this work while he was funded as a Ph.D. student at Université Pierre et Marie Curie in Laboratoire Jacques-Louis Lions. This work was granted access to the HPC resources of TGCC@CEA under the allocations 2017-A0010607519 made by GENCI.

References

[1] P. Sagaut, Large Eddy Simulation for Incompressible Flows, third ed., Springer-Verlag, Berlin, 2006, p. xxx+556.

[2] A.N. Kolmogorov, The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 434 (1890) (1991) 9–13.

[3] J.P. Boris, F.F. Grinstein, E.S. Oran, R.L. Kolbe, New insights into large eddy simulation, Fluid Dyn. Res. 10 (4–6) (1992) 199–228.

[4] T.J.R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Comput. Methods Appl. Mech. Engrg. 127 (1–4) (1995) 387–401.

[5] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method—a paradigm for computational mechanics, Comput. Methods Appl. Mech. Engrg. 166 (1–2) (1998) 3–24.

[6] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Engrg. 197 (1–4) (2007) 173–201.

[7] R. Codina, Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods, Comput. Methods Appl. Mech. Engrg. 190 (13–14) (2000) 1579–1599.

[8] T. Chacón Rebollo, M. Gómez Mármol, V. Girault, I. Sánchez Muñoz, A high order term-by-term stabilization solver for incompressible flow problems, IMA J. Numer. Anal. 33 (3) (2013) 974–1007.

[9] N. Ahmed, T. Chacón Rebollo, V. John, S. Rubino, A review of variational multiscale methods for the simulation of turbulent incompressible flows, Arch. Comput. Methods Eng. 24 (1) (2017) 115–164.

[10] M. Braack, E. Burman, Local projection stabilization for the Oseen problem and its interpretation as a variational multiscale method, SIAM J. Numer. Anal. 43 (6) (2006) 2544–2566.

[11] P. Knobloch, G. Lube, Local projection stabilization for advection-diffusion-reaction problems: one-level vs. two-level approach, Appl. Numer. Math. 59 (12) (2009) 2891–2907.

[12] N. Ahmed, T. Chacón Rebollo, V. John, S. Rubino, Analysis of a full space-time discretization of the Navier–Stokes equations by a local projection stabilization method, IMA J. Numer. Anal. 37 (3) (2017) 1437–1467.

[13] T. Chacón Rebollo, M. Gómez Mármol, M. Restelli, Numerical analysis of penalty stabilized finite element discretizations of evolution Navier-Stokes equation, J. Sci. Comput. 61 (1) (2014) 1–28.

[14] P. Gervasio, F. Saleri, A. Veneziani, Algebraic fractional-step schemes with spectral methods for the incompressible Navier-Stokes equations, J. Comput. Phys. 214 (1) (2006) 347–365.

[15] V. Dolean, P. Jolivet, F. Nataf, An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015, p. x+238.

[16] R. Haferssas, P. Jolivet, F. Nataf, An additive Schwarz method type theory for Lions's algorithm and a symmetrized optimized restricted additive Schwarz method, SIAM J. Sci. Comput. 39 (4) (2017) A1345–A1365.

[17] F. Hecht, New development in FreeFem++, J. Numer. Math. 20 (3–4) (2012) 251–265.

[18] D. Forti, L. Dedè, Semi-implicit BDF time discretization of the Navier-Stokes equations with VMS-LES modeling in a high performance computing framework, Comput. Fluids 117 (2015) 168–182.

[19] O. Colomés, S. Badia, R. Codina, J. Principe, Assessment of variational multiscale models for the large eddy simulation of turbulent incompressible flows, Comput. Methods Appl. Mech. Engrg. 285 (2015) 32–63.

[20] V. Gravemeier, M.W. Gee, M. Kronbichler, W.A. Wall, An algebraic variational multiscale-multigrid method for large eddy simulation of turbulent flow, Comput. Methods Appl. Mech. Engrg. 199 (13–16) (2010) 853–864.

[21] D. Arndt, H. Dallmann, G. Lube, Quasi-optimal error estimates for the incompressible Navier–Stokes problem discretized by finite element methods and pressure-correction projection with velocity stabilization, 2016, https://arxiv.org/abs/1609.00807.

[22] O. Colomés, S. Badia, J. Principe, Mixed finite element methods with convection stabilization for the large eddy simulation of incompressible turbulent flows, Comput. Methods Appl. Mech. Engrg. 304 (2016) 294–318.

[23] C.O. Horgan, Korn's inequalities and their applications in continuum mechanics, SIAM Rev. 37 (4) (1995) 491–511.

[24] T. Chacón Rebollo, R. Lewandowski, Mathematical and Numerical Foundations of Turbulence Models and Applications, Birkhäuser, 2014.

[25] G.P. Galdi, An introduction to the Navier-Stokes initial-boundary value problem, in: Fundamental Directions in Mathematical Fluid Mechanics, in: Adv. Math. Fluid Mech., Birkhäuser, Basel, 2000, pp. 1–70.

[26] R. Becker, M. Braack, A finite element pressure gradient stabilization for the Stokes equations based on local projections, Calcolo 38 (4) (2001) 173–199.

[27] N. Ahmed, G. Matthies, L. Tobiska, H. Xie, Discontinuous Galerkin time stepping with local projection stabilization for transient convection-diffusion-reaction problems, Comput. Methods Appl. Mech. Engrg. 200 (21–22) (2011) 1747–1756.

[28] G.R. Barrenechea, V. John, P. Knobloch, A local projection stabilization finite element method with nonlinear crosswind diffusion for convection-diffusion-reaction equations, ESAIM Math. Model. Numer. Anal. 47 (5) (2013) 1335–1366.

[29] R. Becker, M. Braack, A two-level stabilization scheme for the Navier-Stokes equations, in: Numerical Mathematics and Advanced Applications, Springer-Verlag, 2004, pp. 123–130.

[30] P. Knobloch, A generalization of the local projection stabilization for convection-diffusion-reaction equations, SIAM J. Numer. Anal. 48 (2) (2010) 659–680.

[31] G. Matthies, P. Skrzypacz, L. Tobiska, Stabilization of local projection type applied to convection-diffusion problems with mixed boundary conditions, Electron. Trans. Numer. Anal. 32 (2008) 90–105.

[32] H.-G. Roos, M. Stynes, L. Tobiska, Robust Numerical Methods for Singularly Perturbed Differential Equations. Convection-Diffusion-Reaction and Flow Problems, second ed., in: Springer Series in Computational Mathematics, vol. 24, Springer-Verlag, 2008.

[33] A. Linke, Collision in a cross-shaped domain—a steady 2d Navier-Stokes example demonstrating the importance of mass conservation in CFD, Comput. Methods Appl. Mech. Engrg. 198 (41–44) (2009) 3278–3286.

[34] T. Chacón Rebollo, A term by term stabilization algorithm for finite element solution of incompressible flow problems, Numer. Math. 79 (2) (1998) 283–319.

[35] R. Codina, J. Principe, O. Guasch, S. Badia, Time dependent subscales in the stabilized finite element approximation of incompressible flow problems, Comput. Methods Appl. Mech. Engrg. 196 (21–24) (2007) 2413–2430.

[36] L. He, L. Tobiska, The two-level local projection type stabilization as an enriched one-level approach, Adv. Comput. Math. 36 (4) (2012) 503–523.

[37] L. Tobiska, C. Winkel, The two-level local projection type stabilization as an enriched one-level approach. A one-dimensional study, Int. J. Numer. Anal. Model. 7 (3) (2010) 520–534.

[38] R.L. Scott, S. Zhang, Finite element interpolation of non-smooth functions satisfying boundary conditions, Math. Comp. 54 (190) (1990) 483–493.

[39] S. Badia, On stabilized finite element methods based on the Scott–Zhang projector. Circumventing the inf–sup condition for the Stokes problem, Comput. Methods Appl. Mech. Engrg. 247/248 (2012) 65–72.

[40] M. Olshanskii, G. Lube, T. Heister, J. Löwe, Grad-div stabilization and subgrid pressure models for the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 198 (49–52) (2009) 3975–3988.

[41] R. Codina, J. Blasco, Analysis of a stabilized finite element approximation of the transient convection–diffusion-reaction equation using orthogonal subscales, Comput. Vis. Sci. 4 (3) (2002) 167–174.

[42] R. Codina, A stabilized finite element method for generalized stationary incompressible flows, Comput. Methods Appl. Mech. Engrg. 190 (20–21) (2001) 2681–2706.

[43] T. Chacón Rebollo, F. Hecht, M. Gómez Mármol, G. Orzetti, S. Rubino, Numerical approximation of the Smagorinsky turbulence model applied to the primitive equations of the ocean, Math. Comput. Simulation 99 (2014) 54–70.

[44] T. Chacón Rebollo, M. Gómez Mármol, F. Hecht, S. Rubino, I. Sánchez Muñoz, A high-order local projection stabilization method for natural convection problems, J. Sci. Comput. (2017) 1–26. Published online.

[45] T. Chacón Rebollo, M. Gómez Mármol, S. Rubino, Finite element approximation of an unsteady projection-based VMS turbulence model with wall laws, in: P. Knobloch (Ed.), Boundary and Interior Layers, Computational and Asymptotic Methods - BAIL 2014, in: Lecture Notes in Computational Science and Engineering, vol. 108, Springer, 2015, pp. 47–73.

[46] T. Chacón Rebollo, M. Gómez Mármol, S. Rubino, Numerical analysis of a finite element projection-based VMS turbulence model with wall laws, Comput. Methods Appl. Mech. Engrg. 285 (2015) 379–405.

[47] S. Rubino, Numerical Modeling of Turbulence by Richardson Number-Based and VMS Models (Ph.D. thesis), Univeristy of Seville, 2014.

[48] F.E. Cellier, Continuous System Modeling, Springer-Verlag, New York, 1991, p. xxviii+755.

[49] J.-L. Guermond, Un résultat de convergence d'ordre deux en temps pour l'approximation des équations de Navier-Stokes par une technique de projection incrémentale, ESAIM Math. Model. Numer. Anal. 33 (1) (1999) 169–189.

[50] G. Karypis, V. Kumar, Multilevel *k*-way partitioning scheme for irregular graphs, J. Parallel Distrib. Comput. 48 (1) (1998) 96–129.

[51] X.-C. Cai, M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, SIAM J. Sci. Comput. 21 (2) (1999) 792–797.

[52] V. Dolean, M.J. Gander, L. Gerardo-Giorda, Optimized Schwarz methods for Maxwell's equations, SIAM J. Sci. Comput. 31 (3) (2009) 2193–2213.

[53] P. Jolivet, F. Hecht, F. Nataf, C. Prud'homme, Scalable domain decomposition preconditioners for heterogeneous elliptic problems, in: Proceedings of the 2013 ACM/IEEE Conference on Supercomputing, SC13, ACM, 2013, pp. 80:1–80:11.

[54] F. Auteri, N. Parolini, L. Quartapelle, Numerical investigation on the stability of singular driven cavity flow, J. Comput. Phys. 183 (1) (2002) 1–25.

[55] G. Tiesinga, F.W. Wubs, A.E.P. Veldman, Bifurcation analysis of incompressible flow in a driven cavity by the Newton-Picard method, in: Proceedings of the 9th International Congress on Computational and Applied Mathematics, Leuven, 2000, Vol. 140, 2002, pp. 751–772.

[56] E. Erturk, T.C. Corke, C. Gökçöl, Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers, Int. J. Numer. Methods Fluids 48 (2005) 747–774.

[57] H. Schlichting, K. Gersten, Boundary-Layer Theory, Springer-Verlag, Berlin, 2000, p. xxiv+799.

[58] Y. Zang, R.L. Street, J.R. Koseff, A dynamic mixed subgrid-scale model and its application to turbulent recirculating flows, Phys. Fluids 5 (1993) 3186–3196.

[59] M. Germano, U. Piomelli, P. Moin, W.H. Cabot, A dynamic subgrid-scale eddy viscosity model, Phys. Fluids 3 (1991) 1760–1765.

[60] A.K. Prasad, J.R. Koseff, Reynolds number and end-wall effects on a lid-driven cavity flow, Phys. Fluids 1 (1989) 208–218.

[61] P. Moin, Advances in large eddy simulation methodology for complex flows, Int. J. Numer. Methods Heat Fluid Flow 23 (2002) 710–720.

[62] T. Iliescu, V. John, W.J. Layton, G. Matthies, L. Tobiska, A numerical study of a class of LES models, Int. J. Comput. Fluid Dyn. 17 (1) (2003) 75–85.

[63] J. Smagorinsky, General circulation experiment with the primitive equations: I. The basic experiment, Mon. Weather Rev. 91 (3) (1963) 99–164.

[64] R.A. Clark, J.H. Ferziger, W.C. Reynolds, Evaluation of subgrid-scale models using an accurately simulated turbulent flow, J. Fluid Mech. 91 (1979) 1–16.

[65] G.P. Galdi, W.J. Layton, Approximation of the larger eddies in fluid motions. II. A model for space-filtered flow, Math. Models Methods Appl. Sci. 10 (3) (2000) 343–350.

[66] V. Gravemeier, W.A. Wall, E. Ramm, Large eddy simulation of turbulent incompressible flows by a three-level finite element method, Int. J. Numer. Methods Fluids 48 (10) (2005) 1067–1099.

[67] Intel. MKL web page, 2017. https://software.intel.com/en-us/intel-mkl.

[68] O. Schenk, K. Gärtner, Solving unsymmetric sparse systems of linear equations with PARDISO, Future Gener. Comput. Syst. 20 (3) (2004) 475–487.

[69] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, S. Zampini, H. Zhang, PETSc web page, 2017. http://www.mcs.anl.gov/petsc.

[70] J.-L. Guermond, P. Minev, J. Shen, Error analysis of pressure–correction schemes for the time-dependent Stokes equations with open boundary conditions, SIAM J. Numer. Anal. 43 (1) (2005) 239–258.

[71] A. Poux, S. Glockner, M. Azaïez, Improvements on open and traction boundary conditions for Navier–Stokes time-splitting methods, J. Comput. Phys. 230 (10) (2011) 4011–4027.

[72] A. Linke, G. Matthies, L. Tobiska, Robust arbitrary order mixed finite element methods for the incompressible Stokes equations with pressure independent velocity errors, ESAIM Math. Model. Numer. Anal. 50 (1) (2016) 289–309.