MOX-Report No. 60/2024

Space - time mesh adaptation for the VMS - Smagorinsky modeling of high Reynolds number flows

Temellini, E.; Ferro, N.; Stabile, G.; Delgado Avila, E.; Chacon Rebollo, T.; Perotto, S.

Space-time mesh adaptation for the VMS-Smagorinsky modeling of high Reynolds number flows

E. Temellini[1], N. Ferro[1], G. Stabile[2],
E. Delgado Avila[3], T. Chacon Rebollo[4], S. Perotto[1]

[1] MOX – Modellistica e Calcolo Scientifico, Dipartimento di Matematica
Politecnico di Milano, Piazza L. da Vinci 32, I-20133 Milano, Italy
{`erika.temellini,nicola.ferro,simona.perotto`}`@polimi.it`

[2] Istituto di Biorobotica
Scuola Superiore Sant'Anna, Piazza Martiri della Libertà 33, 56127, Pisa, Italy
{`giovanni.stabile`}`@santannapisa.it`

[3] Department of Quantitave Methods, Universidad Loyola Andalucia
Av. de las Universidades 2, Dos Hermanas, 41704, Spain
`edelgado@uloyola.es`

[4] Ecuaciones Difenrenciales y Analisis Numerico
Universidad de Sevilla,Calle San Fernando 4, 41013, Sevilla, Spain
{`edelgado1, chacon`}`@us.es`

Abstract

Traditional methods, such as Reynolds-Averaged Navier-Stokes (RANS) equations and Large Eddy Simulations (LES), provide consolidated tools for the numerical approximation of high Reynolds number flows in a wide range of applications - from green energy to industrial design. In general, RANS modeling is practical when the main interest is the time-averaged flow behavior. LES equations offer detailed insights into flow dynamics and a more accurate solution, but the high computational demand necessitates innovative strategies to reduce costs while maintaining precision. In this study, we enhance the Variational MultiScale (VMS)-Smagorinsky LES model by relying on an adaptive discretization strategy in both space and time, driven by a recovery-based a posteriori error analysis. We assess the effectiveness of the approach in capturing flow characteristics across a wide range of Reynolds numbers through benchmark tests.

1 Introduction

The numerical approximation of high Reynolds number flows is crucial in various application fields, including green energy technologies and industrial design [1, 2]. These flows, ranging from a laminar to a turbulent regime, are mathematically described by the well-known Navier-Stokes equations [3], possibly supplemented by a turbulence model.

Differently from laminar flows, a turbulent regime exhibits multiscale phenomena, where meaningful dynamics occur across a wide range of scales. The coexistence of different scales make high Reynolds flow simulations very demanding in terms of computational resources, when resorting to a direct numerical simulation (DNS) approach. Indeed, the accurate resolution of the involved scales requires a sufficiently fine spatial grid spacing, strictly related to a certain power of the Reynolds number [4]. In addition, turbulent flows lead to deal with time-dependent phenomena, which may exhibit strong heterogeneities in the whole time window. Thus, the DNS modeling of turbulent flows, in particular in a 3D setting and in long time-frame, unavoidably demands for a prohibitive computational effort.

As a remedy to DNS, classical approaches are used in the literature, namely Reynolds-Averaged Navier-Stokes (RANS) equations and Large Eddy Simulations (LES) [5, 6, 7]. RANS equations are generally used to describe the mean flow behaviour, since derived by averaging the Navier-Stokes equations in time, making them well-suited to study time-averaged effects like forces on structures. On the contrary, when the focus is on detailed flow dynamics, it is customary to use LES since they resolve the large scales of the model and models the effects of the smaller scales, through a filtering approach. For instance, the Smagorinsky LES model takes into account the effect of the unresolved scales by a turbulence viscosity correction [8].

As a result, RANS equations are characterized by an affordable computational cost but a low accuracy. Conversely, LES equations entail higher computational costs and greater precision. This issue has motivated research efforts in the literature to develop new cutting-edge models and methods that balance computational efficiency with fidelity (see, e.g., [9, 10, 11, 12]).

More recently, the Variational Multiscale (VMS) methodology offered a viable alternative to traditional LES, by distinguishing the modeled scales into two or three classes [13]. This approach proved to be effective, in particular when modelling high Reynolds number flows [14, 11, 15]. In this paper, we focus on the VMS-Smagorinsky model introduced in [16], where a three-scale VMS formulation is completed by the Smagorinsky turbulence correction in order to localize the viscosity effect onto specific subscales. To balance the computational effort due to the modeling of several scales, we settle a procedure for an automatic choice of the discretization in order to match the flow heterogeneities both in space and time. To this aim, we resort to an adaptive selection of the spatial computational mesh as well as of the time-step, driven by a space-time a posteriori recovery-based error analysis. The discretization of the spatial domain is carried out in an anisotropic setting [17, 18] with the aim to sharply capture velocity variation, for instance, near the domain boundaries or corners [19, 20, 21, 22]. The time window is discretized with a dual purpose: to accurately track the fast dynamics using a small time-step, while relaxing this constraint when the flow reaches a steady regime [23, 24]. To the best of our knowledge, this study represents the first attempt where an anisotropic spatial adaptive procedure is enhanced by an adaptive selection of the time-step driven by a recovery-based error estimator, for the modeling of high Reynolds number flows.

The paper is organized as follows. In Sect. 2, we present the VMS-Smagorinsky model for the Navier-Stokes equations. Section 3 settles the proposed adaptive methodology. In particular, the space and time adaptations are first managed separately and, then, combined to deliver the whole adaptive process. In Sect. 4 we assess different adaptive procedures on benchmark case studies.We challenge the proposed approach across different Reynolds numbers, thus ranging from laminar to turbulent regimes, and we verify the obtained outcomes with respect to reference results in the literature. In the last section, some conclusions are drawn and future perspectives are provided.

2 The reference turbulence context

The resolution of all the scales involved in turbulence phenomena leads, in general, to a computionally unaffordable modeling, as previously mentioned. For this reason, Large Eddy Simulations (LES) equations resolve the fluid dynamics up to a predefined length scale. In particular, LES equations model the effect of the so-called sub-scales by adding a correction term to the classical Navier-Stokes (NS) equations. This additional contribution, represented by an artificial viscosity, takes into account the dissipative effect of the smallest scales. The most widely employed LES model is the Smagorinsky turbulence correction firstly presented in [8].

Over the past 30 years, Variational Multiscale (VMS) methodology has emerged as a viable alternative to the traditional LES approach, by introducing a further distinction in the modeled scales into two or three classes, see [13], VMS proved to be particularly effective when modeling high Reynolds number flows, namely when accuracy and detailed resolution are needed.

In this work, we resort to the VMS-Smagorinsky model presented in [16], where the Smagorinsky artificial viscosity correction is applied only to one of three scales highlighted by the VMS method. The VMS-Smagorinsky approach tends to dampen the over diffusive effect characterizing the classical Smagorinsky model, also leading to computational savings.

2.1 The Navier-Stokes equations

The NS equations, based on the principles of mass and momentum conservation, and of energy balance, provide a comprehensive framework for modeling the fluid flow dynamics under various physical conditions, see [3]. In the following, we consider an incompressible flow in the space-time cylinder $\mathcal{Q} = \Omega \times I$, with $\Omega \subset \mathbb{R}^2$ the spatial domain of boundary $\partial\Omega$, and $I = (t_0, t_f]$ the time window. In particular, we look for the fluid velocity

$\mathbf{u} = [u_x(\mathbf{x},t), u_y(\mathbf{x},t)]^T$, and the kinetic pressure, $p = p(\mathbf{x},t)$, with $\mathbf{x} = (x,y)$, such that

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t}(\mathbf{x},t) + (\mathbf{u}(\mathbf{x},t)\cdot\nabla)\,\mathbf{u}(\mathbf{x},t) - \nu\Delta\mathbf{u}(\mathbf{x},t) + \dfrac{1}{\rho}\nabla p(\mathbf{x},t) = \mathbf{f}(\mathbf{x},t) & \text{in } \mathcal{Q} \\ \nabla\cdot\mathbf{u}(\mathbf{x},t) = \mathbf{0} & \text{in } \mathcal{Q} \\ \mathbf{u}(\mathbf{x},t) = \boldsymbol{\varphi}(\mathbf{x},t) & \text{on } \Gamma_D \times I \\ \nu\dfrac{\partial \mathbf{u}}{\partial n}(\mathbf{x},t) - \dfrac{1}{\rho}\, p(\mathbf{x},t)\mathbb{I}\mathbf{n} = \boldsymbol{\psi}(\mathbf{x},t) & \text{on } \Gamma_N \times I \\ \mathbf{u}(\mathbf{x},t_0) = \mathbf{u}_0(\mathbf{x}) & \text{in } \Omega, \end{cases} \tag{1}$$

where ν and ρ are the kinematic viscosity and the fluid density, respectively, here assumed to be constant and positive, $\mathbf{f}$ is the force per unit mass, $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ impose the flow behaviour at the boundaries Γ_D and Γ_N, respectively, with $\Gamma_D \cup \Gamma_N = \partial\Omega$, $\mathring{\Gamma}_D \cap \mathring{\Gamma}_N = \emptyset$, $\mathbb{I} \in \mathbb{R}^{2\times 2}$ is the identity matrix, $\mathbf{n}$ is the unit outward vector normal to the boundary, and $\mathbf{u}_0$ is the velocity profile at $t = t_0$. Throughout the paper, boundary data $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ are chosen constant in time. Moreover, we omit the dependence on $\mathbf{x}$ and t to simplify the notation.

With a view to the approximation, we consider the weak formulation of problem (1) that reads as follows: $\forall t \in I$, find $\mathbf{u} = \mathbf{u}(t) \in Y = [H^1(\Omega)]^2$ and $p = p(t) \in M = L^2(\Omega)$ such that, for any $\mathbf{v} \in Y$ and $q \in M$,

$$\begin{aligned} &\int_\Omega \left\{ \frac{\partial \mathbf{u}}{\partial t}\cdot\mathbf{v} + [(\mathbf{u}\cdot\nabla)\,\mathbf{u}]\cdot\mathbf{v} + \nu\nabla\mathbf{u} : \nabla\mathbf{v} - \frac{1}{\rho}\, p\nabla\cdot\mathbf{v} - q\nabla\cdot\mathbf{u} \right\} d\Omega \\ = &\int_\Omega \mathbf{f}\cdot\mathbf{v}\, d\Omega + \int_{\Gamma_N} \boldsymbol{\psi}\cdot\mathbf{v}\, d\Gamma, \end{aligned} \tag{2}$$

with $\mathbf{u}|_{\Gamma_D} = \boldsymbol{\varphi}$, $\mathbf{u}(\mathbf{x},t_0) = \mathbf{u}_0$, and where we adopt standard notation for the function spaces, see [25].

In order to approximate problem (2), we discretize the cylinder $\mathcal{Q}$ by introducing the spatial triangular tessellation, $\mathcal{T}_h = \{K\}$, of the domain Ω of size $h = \max_K \operatorname{diam}(K)$, and the partition, $\{t^0, t^1, \ldots, t^f\}$, of the time window of length $\Delta t = \max_{k=0,\ldots f-1}(t^{k+1} - t^k)$, with $I_k = [t^k, t^{k+1}]$ the generic subinterval for $k = 0, ..., f-1$. In particular, we adopt a finite element scheme for the spatial discretization, combined with a semi-implicit time advancing method [25], so that we have: for $n = 0, ..., f-1$, find $\mathbf{u}_h^{n+1} = \mathbf{u}_h(t^{n+1}) = [u_{h,x}^{n+1}, u_{h,y}^{n+1}]^T \in Y_h \subset Y$ and $p_h^{n+1} = p_h(t^{n+1}) \in M_h \subset M$, with $\dim(Y_h)$, $\dim(M_h) < +\infty$, such that, for any $\mathbf{v}_h \in Y_h$ and $q_h \in M_h$,

$$\begin{aligned} &\int_\Omega \Big\{ \frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\Delta t}\cdot\mathbf{v}_h + [(\mathbf{u}_h^n\cdot\nabla)\mathbf{u}_h^{n+1}]\cdot\mathbf{v}_h + \nu\nabla\mathbf{u}_h^{n+1} : \nabla\mathbf{v}_h \\ - &\ \frac{1}{\rho} p_h^{n+1}\nabla\cdot\mathbf{v}_h - q_h\nabla\cdot\mathbf{u}_h^{n+1} \Big\} d\Omega = \int_\Omega \mathbf{f}^{n+1}\cdot\mathbf{v}_h\, d\Omega + \int_{\Gamma_N} \boldsymbol{\psi}_h\cdot\mathbf{v}_h\, d\Gamma, \end{aligned} \tag{3}$$

with $\mathbf{u}_h^{n+1}|_{\Gamma_D} = \boldsymbol{\varphi}_h$, $\mathbf{u}_h^0 = \mathbf{u}_{h,0}$, and $\boldsymbol{\varphi}_h$, $\boldsymbol{\psi}_h$ and $\mathbf{u}_{h,0}$ suitable approximations of $\boldsymbol{\varphi}$, $\boldsymbol{\psi}$ and $\mathbf{u}_0$ in Y_h, respectively.
With a view to the VMS-Smagorinsky formulation, we rewrite (3) in a compact way as

$$\begin{aligned}&\mathcal{A}_t(\mathbf{u}_h^{n+1},\mathbf{v}_h)+\mathcal{A}_c(\mathbf{u}_h^{n+1},\mathbf{v}_h;\mathbf{u}_h^n)+\mathcal{A}_d(\mathbf{u}_h^{n+1},\mathbf{v}_h)\\ +\ &\mathcal{A}_p(p_h^{n+1},\mathbf{v}_h)+\mathcal{A}_m(q_h,\mathbf{u}_h^{n+1})=\mathcal{F}(\mathbf{f}^{n+1},\mathbf{v}_h)+\mathcal{B}(\boldsymbol{\psi}_h,\mathbf{v}_h)\end{aligned}\tag{4}$$

where

$$\mathcal{A}_t(\mathbf{u}_h^{n+1},\mathbf{v}_h)=\int_\Omega\frac{\mathbf{u}_h^{n+1}-\mathbf{u}_h^n}{\Delta t}\cdot\mathbf{v}_h\,d\Omega,\quad \mathcal{A}_c(\mathbf{u}_h^{n+1},\mathbf{v}_h;\mathbf{u}_h^n)=\int_\Omega[(\mathbf{u}_h^n\cdot\nabla)\mathbf{u}_h^{n+1}]\cdot\mathbf{v}_h\,d\Omega,$$

$$\mathcal{A}_d(\mathbf{u}_h^{n+1},\mathbf{v}_h)=\int_\Omega\nu\nabla\mathbf{u}_h^{n+1}:\nabla\mathbf{v}_h\,d\Omega,\quad \mathcal{A}_p(p_h^{n+1},\mathbf{v}_h)=-\int_\Omega\frac{1}{\rho}p_h^{n+1}\nabla\cdot\mathbf{v}_h\,d\Omega,$$

$$\mathcal{A}_m(q_h,\mathbf{u}_h^{n+1})=-\int_\Omega q_h\nabla\cdot\mathbf{u}_h^{n+1}\,d\Omega,\qquad \mathcal{F}(\mathbf{f}^{n+1},\mathbf{v}_h)=\int_\Omega\mathbf{f}^{n+1}\cdot\mathbf{v}_h\,d\Omega,$$

$$\mathcal{B}(\boldsymbol{\psi}_h,\mathbf{v}_h)=\int_{\Gamma_N}\boldsymbol{\psi}_h\cdot\mathbf{v}_h\,d\Gamma.$$

Concerning the discrete spaces Y_h and M_h, we select $Y_h=[X_h^2]^2$ and $M_h=X_h^1$, with

$$X_h^r=\left\{w_h\in C^0(\bar{\Omega})\,:\,w_h|_K\in\mathbb{P}_r(K),\ \forall K\in\mathcal{T}_h\right\},$$

namely quadratic and affine finite elements to approximate the velocity and the pressure, respectively, thus ensuring the inf-sup condition, see [26]. In addition, the space size h and the time length Δt have to be properly related in order to match the CFL condition, see [27].

2.2 Smagorinsky turbulence model

The LES approach separates the flow into length scales above and below a certain threshold δ. In particular, the former are explicitly solved, while the latter (known as subscales) are modelled by adding a correction term to the NS equations.

From a physical viewpoint, the subscales are responsible for a dissipative effect onto the larger scales so that it is customary to identify the LES correction with an additive artificial viscosity ν_T. According to the definition of ν_T, we distinguish different LES variants. In this work, we focus on the Smagorinsky model for an incompressible viscous flow. In [8], based on a dimensional analysis, J. Smagorinsky defined the artificial viscosity as

$$\nu_T=(C_s\delta)^2\,|\nabla\mathbf{u}|\,,\tag{5}$$

where C_s is the Smagorinsky coefficient – here set to 0.18 according to the literature, see for instance [28] – and $|\nabla\mathbf{u}|$ is the strain rate tensor modulus.

In a computational perspective, the length scale δ is replaced by the mesh size h, identifying the minimum length scale that can be solved in practice, so that the subscales are known as subgrid scales . Thus, the NS equations endowed with the Smagorinsky turbulence model are provided by: for $n=0,...,f-1$, find $\mathbf{u}_h^{n+1}\in Y_h$ and $p_h^{n+1}\in M_h$, such that, for any $\mathbf{v}_h \in Y_h$ and $q_h \in M_h$,

$$\begin{aligned}
&\mathcal{A}_t(\mathbf{u}_h^{n+1}, \mathbf{v}_h) + \mathcal{A}_c(\mathbf{u}_h^{n+1}, \mathbf{v}_h; \mathbf{u}_h^n) + \mathcal{A}_d(\mathbf{u}_h^{n+1}, \mathbf{v}_h) + \mathcal{A}_S(\mathbf{u}_h^{n+1}, \mathbf{v}_h; \mathbf{u}_h^n) \\
+ \quad &\mathcal{A}_p(p_h^{n+1}, \mathbf{v}_h) + \mathcal{A}_m(q_h, \mathbf{u}_h^{n+1}) = \mathcal{F}(\mathbf{f}^{n+1}, \mathbf{v}_h) + \mathcal{B}(\boldsymbol{\psi}_h, \mathbf{v}_h)
\end{aligned} \tag{6}$$

with $\mathbf{u}_h^{n+1}|_{\Gamma_D} = \boldsymbol{\varphi}_h$, $\mathbf{u}_h^0 = \mathbf{u}_{h,0}$, where

$$\mathcal{A}_S(\mathbf{u}_h^{n+1}, \mathbf{v}_h; \mathbf{u}_h^n) = \int_\Omega (C_s h)^2 \, |\nabla \mathbf{u}_h^n| \, \nabla \mathbf{u}_h^{n+1} : \nabla \mathbf{v}_h \, d\Omega. \tag{7}$$

The Smagorinsky correction is treated in a semi-implcit way, analogously to the convective contribution in the NS model.

2.3 VMS-Smagorinsky approach

The classical Smagorinsky model in (6) is known to exhibit an over-diffusive effect [5]. The issue is that the artificial viscosity ν_T is applied uniformly across the entire domain. However, from a physical perspective, the dissipation introduced by the subgrid scales (SGS) does not affect the largest scales but primarily impacts the medium-sized scales. The VMS-Smagorinsky formulation addresses this over-diffusive effect by applying the Smagorinsky artificial viscosity only to the medium-sized scales, after separating the flow field $\mathbf{u}_h$ into

- large resolved scales ($\bar{\mathbf{u}}_h$), whose size and shape are influenced by the geometry of the problem, and that are unaffected by the dissipative effects of the SGS;
- small resolved scales ($\hat{\mathbf{u}}_h$), i.e. the medium-size scales, coinciding with the smallest ones to be numerically solved and that are affected by the dissipative effects of the SGS;
- small unresolved scales (coinciding with the SGS), i.e., the smallest scales that are not numerically solved but whose presence is kept into account through the Smagorinsky artificial viscosity ν_T,

according to a three-scale VMS method [15, 11]. Here, we denote by $\bar{\mathbf{u}}_h \in \bar{Y}_h \subset Y_h$ the large resolved velocity and by $\hat{\mathbf{u}}_h \in \hat{Y}_h \subset Y_h$ the corresponding medium-sized velocity field, so that

$$\mathbf{u}_h = (\bar{\mathbf{u}}_h + \hat{\mathbf{u}}_h) \in Y_h = \bar{Y}_h \oplus \hat{Y}_h.$$

Thus, the classical Smagorinsky correction $\mathcal{A}_S\left(\mathbf{u}_h^{n+1}, \mathbf{v}_h; \mathbf{u}_h^n\right)$ in (7) is replaced by the VMS-Smagorinsky eddy viscosity contribution, given by

$$\mathcal{A}_{VS}\left(\mathbf{u}_h^{n+1}, \mathbf{v}_h; \mathbf{u}_h^n\right) = \int_\Omega (C_s h)^2 \, |\nabla \hat{\mathbf{u}}_h^n| \, \nabla \hat{\mathbf{u}}_h^{n+1} : \nabla \hat{\mathbf{v}}_h \, d\Omega, \tag{8}$$

after setting

$$\hat{\mathbf{u}}_h^\gamma = (\mathcal{I} - \Pi_h)\, \mathbf{u}_h^\gamma, \qquad \hat{\mathbf{v}}_h = (\mathcal{I} - \Pi_h)\, \mathbf{v}_h, \tag{9}$$

with $\gamma = n, n+1$, and where $\mathcal{I}$ is the identity operator in Y_h, while $\Pi_h : Y_h \to \bar{Y}_h$ is the projection operator that extracts the large resolved scales. Relation (8) confirms that VMS-Smagorinsky method indeed confines the diffusive action of SGS to the middle-sized scales.

The VMS-Smagorinsky discrete problem involved in the numerical validation in Section 4 can thus be stated: for $n = 0, ..., f-1$, find $\mathbf{u}_h^{n+1} \in Y_h$ and $p_h^{n+1} \in M_h$, such that, for any $\mathbf{v}_h \in Y_h$ and $q_h \in M_h$,

$$\begin{aligned} &\mathcal{A}_t(\mathbf{u}_h^{n+1}, \mathbf{v}_h) + \mathcal{A}_c(\mathbf{u}_h^{n+1}, \mathbf{v}_h; \mathbf{u}_h^n) + \mathcal{A}_d(\mathbf{u}_h^{n+1}, \mathbf{v}_h) + \mathcal{A}_{VS}(\mathbf{u}_h^{n+1}, \mathbf{v}_h; \mathbf{u}_h^n) \\ + \quad &\mathcal{A}_p(p_h^{n+1}, \mathbf{v}_h) + \mathcal{A}_m(q_h, \mathbf{u}_h^{n+1}) = \mathcal{F}(\mathbf{f}^{n+1}, \mathbf{v}_h) + \mathcal{B}(\boldsymbol{\psi}_h, \mathbf{v}_h) \end{aligned} \tag{10}$$

with $\mathbf{u}_h^{n+1}|_{\Gamma_D} = \boldsymbol{\varphi}_h$, $\mathbf{u}_h^0 = \mathbf{u}_{h,0}$ (for the theoretical analysis of this model see [16]). In particular, concerning the operator Π_h in (9), we opt for $\Pi_h : Y_h \to Y_h$, with $\Pi_h = \Pi_h^{1\to 2} \circ \Pi_h^{2\to 1}$, being $\Pi_h^{1\to 2} : [X_h^1]^2 \to Y_h$ and $\Pi_h^{2\to 1} : Y_h \to [X_h^1]^2$ standard projection operators between finite element spaces.

3 A recovered error-based space-time adaptive discretization

Handling high Reynolds number flows requires a fine space and time discretization to capture flow features accurately, leading to a high number of degrees of freedom and making simulations computationally expensive. To address this numerical issue, we resort to an adaptive space-time discretization, which guarantees an accurate modeling at an affordable computational cost. This goal can be achieved by strategically balancing the discretization error across the space-time domain, as already done in [29, 30, 31, 32, 33]. Typically, adaptive procedures rely on heuristic or mathematically rigorous error quantifiers, i.e., the so-called error indicators or estimators [34, 35, 36].

In this work, we adopt a recovery-based error analysis to customize the space and time discretization to the solution of problem (10). Recovery-based error estimators provide an a posteriori (namely, a computable) quantification for the H^1-seminorm of the discretization error associated with the numerical solution to a generic PDE problem. In more detail, we denote by $z = z(\mathbf{x}, t) \in Z$ such a solution and by $z_h = z_h(\mathbf{x}, t) \in Z_h \subset Z$ the corresponding approximation yielded by a certain numerical scheme (not necessarily, a finite element function), with Z and Z_h suitable continuous and discrete function spaces. Following the seminal papers by O.C. Zienckiewicz and J.Z. Zhu [37, 38, 39], the seminorm

$$|e_z|^2_{H^1(\mathcal{D})} = \int_{\mathcal{D}} |\Theta z - \Theta z_h|^2 \, d\mathcal{D}, \tag{11}$$

of the discretization error $e_z = z - z_h$, is made computable by replacing Θz – with Θ a first order partial derivative when $\mathcal{D} \subset \mathbb{R}$, $\Theta \equiv \nabla$ when $\mathcal{D} \subset \mathbb{R}^2$ – with the recovered quantity $\mathcal{R}_\Theta(z_h)$, so that

$$|e_z|^2_{H^1(\mathcal{D})} \simeq \int_{\mathcal{D}} |\mathcal{R}_\Theta(z_h) - \Theta z_h|^2 \, d\mathcal{D} = [\eta^z_{\Theta,\mathcal{D}}]^2. \tag{12}$$

In particular, $\mathcal{R}_\Theta(z_h)$ denotes the recovered derivative of z_h, which, in general, provides a better approximation of Θz with respect to Θz_h. Several recipes for $\mathcal{R}_\Theta(z_h)$ are available in the literature, customized to the specific setting at hand [40, 41, 42, 43]. The estimators as in (12) are widely employed in practical engineering problems motivated by the remarkable effectiveness, the independence of the specific problem and of the adopted discretization, the handy implementation, and the computational cheapness.

Estimator $\eta^z_{\Theta,\mathcal{D}}$ will be employed in the following to carry out the adaptation of the space and the time discretization of the domain $\mathcal{Q}$ in (1), by properly selecting the driving quantity z, the differential operator Θ and the domain $\mathcal{D}$. After managing the two adaptation strategies, separately (see Sect. 3.1-3.2), we will set a unified environment to grasp the benefits of the two procedures in a combined manner (see Sect. 3.3).

3.1 Adaptation in space: an anisotropic setting

To perform the adaptation of the spatial mesh, $\mathcal{T}_h^n$, at time t^n, we exploit the estimator in (12) for $\Theta = \nabla$, $\mathcal{D} = \Omega$, $z_h = U_h^n$ a scalar function at t^n, namely

$$\begin{aligned} \left[\eta^U_{\nabla,\Omega}(t^n)\right]^2 &= \int_\Omega |\mathcal{R}_\nabla(U_h^n) - \nabla U_h^n|^2 \, d\Omega \\ &= \sum_{K\in\mathcal{T}_h^n} \int_K |\mathcal{R}_\nabla(U_h^n) - \nabla U_h^n|^2 \, dK = \sum_{K\in\mathcal{T}_h^n} \left[\eta^U_{\nabla,\Omega}(t^n)\right]^2_K, \end{aligned} \tag{13}$$

where $\left[\eta^U_{\nabla,\Omega}(t^n)\right]_K$ denotes the local (i.e., elementwise) error estimator, with $\mathcal{R}_\nabla(U_h^n) = |\Delta_K|^{-1}\sum_{T\in\Delta_K} |T| \; \nabla U_h^n|_T$ the area-weighted average of the discrete gradient over the patch $\Delta_K = \{T \in \mathcal{T}_h^n : T \cap K \neq \emptyset\}$ of the elements associated with K, see [40], and $|\omega|$ the area of the open set $\omega \subset \Omega$. In particular, we resort to the anisotropic variant of the estimator in (13), see [44], where

$$\begin{aligned} \left[\tilde\eta^U_{\nabla,\Omega}(t^n)\right]^2 &= \sum_{K\in\mathcal{T}_h^n} \left[\tilde\eta^U_{\nabla,\Omega}(t^n)\right]^2_K \\ &= \sum_{K\in\mathcal{T}_h^n} \frac{1}{\lambda^n_{1,K}\lambda^n_{2,K}} \sum_{i=1}^2 (\lambda^n_{i,K})^2 \left((\mathbf{r}^n_{i,K})^T \; G_K(\mathcal{R}_\nabla(U_h^n) - \nabla U_h^n) \; \mathbf{r}^n_{i,K}\right). \end{aligned} \tag{14}$$

Following [17], the geometric quantities $\{\lambda^n_{i,K}\}_{i=1}^2$ and $\{\mathbf{r}^n_{i,K}\}_{i=1}^2$ characterize the anisotropic features of the generic triangle, identifying the semi-lengths and the directions of the axes of the ellipse circumscribed to K according to a standard finite element setting [25]. Concerning G_K, it is the symmetric semi-definite positive matrix with entries $[G_K(\mathbf{w})]_{ij} = \int_{\Delta_K} w_i w_j \, d\Delta_K$ for $i, j = 1, 2$ and $\mathbf{w} = [w_1, w_2]^T \in [L^2(\Omega)]^2$ a generic vector-valued function, which is employed to project the information associated with $\eta^U_{\nabla,\Omega}(t^n)$ along the anisotropic directions $\mathbf{r}_{i,K}$.

The main motivation behind the choice for an anisotropic mesh adaptation is the well-recognized computational gain when compared to an adapted isotropic tessellation, see, for instance, [45, 46]. Indeed, anisotropic meshes accurately track the solution features by tuning the element size, shape, and orientation, in contrast to an isotropic context where only the triangle size can be adjusted. This justifies the large employment of anisotropic adapted meshes in common computational tools, in particular for the modeling of markedly directional phenomena. In particular, estimator (14) has been successfully used in several application fields [23, 24, 47, 48, 49, 50].

The anisotropic error estimator (14) is exploited to a drive a mesh adaptation procedure. We resort to a metric-based approach [51]. In this context, estimator $\tilde{\eta}^U_{\nabla,\Omega}(t^n)$ is utilized to predict the spacing associated with the new adapted mesh $\mathcal{T}_h^{n,*}$ at time t^n. The new element distribution is provided through the piecewise constant symmetric positive tensor field, $M^{n,*} = M^{n,*}(\{\lambda_{i,K}^{n,*}, \boldsymbol{r}_{i,K}^{n,*}\}_{i=1}^2) \in \mathbb{R}^{2\times 2}$, such that, for any $K \in \mathcal{T}_h^{n,*}$,

$$M^{n,*}\big|_K = (R_K^{n,*})^T (\Lambda_K^{n,*})^{-2}\, R_K^{n,*} \text{ with } R_K^{n,*} = [\boldsymbol{r}_{1,K}^{n,*}, \boldsymbol{r}_{2,K}^{n,*}]^T,\ \Lambda_K^{n,*} = \mathrm{diag}(\lambda_{1,K}^{n,*}, \lambda_{2,K}^{n,*}), \tag{15}$$

where $\lambda_{i,K}^{n,*}$ and $\boldsymbol{r}_{i,K}^{n,*}$ preserve the same geometric meaning as quantities $\lambda_{i,K}^{n}$ and $\boldsymbol{r}_{i,K}^{n}$ in (14).

Metric $M^{n,*}$ is generated according to specific criteria, namely, in order to i) ensure a certain accuracy, τ_s, on the error estimator; ii) minimize the mesh cardinality, $\#\mathcal{T}_h^{n,*}$; iii) equidistribute the error across the mesh elements, i.e., $\big[\tilde{\eta}^U_{\nabla,\Omega}(t^n)\big]_K^2 = \tau_s^2/\#\mathcal{T}_h^n$. These requests lead to solve elementwise the constrained minimization problem

$$[s_K^{n,*}, \boldsymbol{r}_{i,K}^{n,*}] = \operatorname*{argmin}_{s_K^n, \boldsymbol{r}_{i,K}^n} \mathcal{J}\big(s_K^n, \boldsymbol{r}_{1,K}^n, \boldsymbol{r}_{2,K}^n\big) \ \text{ such that } \ s_K^n \geq 1,\ \boldsymbol{r}_{i,K}^n \cdot \boldsymbol{r}_{j,K}^n = \delta_{i,j},\ i,j = 1,2, \tag{16}$$

with $\delta_{i,j}$ the Kronecker symbol, $s_K^n = \lambda_{1,K}^n/\lambda_{2,K}^n$ the aspect ratio of the element $K \in \mathcal{T}_h^n$,

$$\begin{aligned}\mathcal{J}\big(s_K^n, \boldsymbol{r}_{1,K}^n, \boldsymbol{r}_{2,K}^n\big) &= s_K^n \Big((\boldsymbol{r}_{1,K}^n)^T \widehat{G}_{\Delta_K}(\mathcal{R}_\nabla(U_h^n) - \nabla U_h^n)\, \boldsymbol{r}_{1,K}^n \Big) \\ &+ (s_K^n)^{-1} \Big((\boldsymbol{r}_{2,K}^n)^T \widehat{G}_{\Delta_K}(\mathcal{R}_\nabla(U_h^n) - \nabla U_h^n)\, \boldsymbol{r}_{2,K}^n \Big),\end{aligned}$$

$\widehat{G}_{\Delta_K}(\cdot) = G_{\Delta_K}(\cdot)/\big|\Delta_K\big|$, after rewriting the local error estimator (14) as

$$\big[\tilde{\eta}^U_{\nabla,\Omega}(t^n)\big]_K^2 = \lambda_{1,K}^n \lambda_{2,K}^n \widehat{C}_K \mathcal{J}\big(s_K^n, \boldsymbol{r}_{1,K}^n, \boldsymbol{r}_{2,K}^n\big), \tag{17}$$

in order to single out the area information, being $\big|\Delta_K\big| = \lambda_{1,K}\lambda_{2,K}\widehat{C}_K$, with $\widehat{C}_K$ an explicitly computable constant, which depends on the finite element reference triangle (see [25] for more details). Notice that the algebraic manipulation converting (14) into (17) exploits the equivalence between the maximization of the element area and the minimization of the mesh cardinality in ii).

The solution to the minimization problem (16) can be computed as

$$s_K^{n,*} = \sqrt{g_1/g_2}, \quad \boldsymbol{r}_{1,K}^{n,*} = \boldsymbol{g}_2, \quad \boldsymbol{r}_{2,K}^{n,*} = \boldsymbol{g}_1, \tag{18}$$

with $\{g_i, \boldsymbol{g}_i\}_{i=1}^2$ the eigen-pairs associated with the matrix $\widehat{G}_{\Delta_K}(\mathcal{R}_\nabla(U_h^n) - \nabla U_h^n)$. Finally, the optimal values $\lambda_{i,K}^{n,*}$ are derived through criterion iii), yielding

$$\lambda_{1,K}^{n,*} = g_2^{-1/2} \left(\frac{\tau_s^2}{2\,\#\mathcal{T}_h^n\, \widehat{C}_K} \right)^{1/2}, \quad \lambda_{2,K}^{n,*} = g_1^{-1/2} \left(\frac{\tau_s^2}{2\,\#\mathcal{T}_h^n\, \widehat{C}_K} \right)^{1/2}. \tag{19}$$

Quantities $\boldsymbol{r}_{i,K}^{n,*}$ and $\lambda_{i,K}^{n,*}$ in (18)-(19) identify the new metric $M^{n,*}$ prompting the generation of $\mathcal{T}_h^{n,*}$ (see [52] for the details).

3.2 Adaptation in time

The adaptive procedure in time that we adopt is meant to predict, at a certain time t^n, the time-step $\Delta t^{n+1} = t^{n+1} - t^n$, namely the length of the subsequent time interval $I_{n+1} = [t^n, t^{n+1}]$. This means that we do not know a priori the number of time intervals spanning the time window $[t_0, t_f]$. We again exploit the estimator in (12) by selecting Θ as a backward finite difference approximation, δ_-, of the first order partial derivative in time, $\mathcal{D} = I_n$, and $z_h = U_h^V$, the same quantity as in Sect. 3.1 now evaluated at a fixed vertex $V \in \mathcal{T}_h^n$. Thus, the recovery-based a posteriori temporal error estimator at V reads as

$$\left[\eta_{\delta_-,I_n}^U(V)\right]^2 = \int_{I_n} \left| \mathcal{R}_{\delta_-}(U_h^V) - \frac{U_h^V(t^n) - U_h^V(t^{n-1})}{\Delta t^n} \right|^2 dt, \tag{20}$$

where

$$\mathcal{R}_{\delta_-}(U_h^V) = \frac{\partial \mathcal{U}_h^V(t)}{\partial t},$$

with $\mathcal{U}_h^V(t)$ the quadratic polynomial interpolating the pairs of data $(t^{n-2}, U_h^V(t^{n-2}))$, $(t^{n-1}, U_h^V(t^{n-1}))$, $(t^n, U_h^V(t^n))$ (see, for instance, [23]). To remove the spatial information from $\left[\eta_{\delta_-,I_n}^U(V)\right]$, we average the vertex information throughout the mesh elements, so that the global error estimator associated with I_n becomes

$$\left[\eta_{\delta_-,I_n}^U\right]^2 = \frac{1}{3} \sum_{K \in \mathcal{T}_h^n} \sum_{V \in K} \left[\eta_{\delta_-,I_n}^U(V)\right]^2. \tag{21}$$

To make this quantity useful to predict the time-step Δt^{n+1}, we introduce a suitable rescaling and adimensionalization, so that we define

$$\left[\tilde{\eta}_{\delta_-,I_n}^U\right]^2 = \tilde{T}\left[\Delta t^n \rho_{\delta_-,I_n}^U\right]^2 \tag{22}$$

with $\tilde{T}$ a suitable time scale factor characteristic of the problem at hand (see [23] for further details), and

$$\left[\rho_{\delta_-,I_n}^U\right]^2 = \frac{1}{3\left[\Delta t^n\right]^2} \sum_{K \in \mathcal{T}_h^n} \sum_{V \in K} \left[\eta_{\delta_-,I_n}^U(V)\right]^2.$$

Since the actual number of time subintervals is not known a priori, we set the same tolerance, τ_t, for each subinterval I_n, i.e., we impose

$$\tilde{T}\left[\Delta t^n \rho_{\delta_-,I_n}^U\right]^2 = \tau_t^2, \tag{23}$$

thus enforcing the error equidistribution throughout the time window. Finally, relation (23) is exploited in a predictive manner (i.e., to compute Δt^{n+1}), yielding

$$\Delta t^{n+1} = \sqrt{\frac{1}{\tilde{T}}} \frac{\tau_t}{\rho_{\delta_-,I_n}^U}. \tag{24}$$

3.3 Space-time adaptation

To efficiently combine the adaption in space and time in Sects. 3.1-3.2, we adopt the procedure sketched in Fig. 1. At time t^n, we compute the solution to problem (10) resorting to a space-time discretization characterized by mesh $\mathcal{T}_h^n$ and time-step Δt^n. Successively, we evaluate the space and time estimators in (14) and (22), from now on denoted by $\eta_s(t^n)$ and $\eta_t(t^n)$ to simplify the notation. Now, a priori, it is possible to resort to both spatial and temporal grid adaptation at each time instant. In practice, since the problem solution can alternate abrupt changes and steady states, both in space and time, we execute the adaptation module only when the variation of η_s and η_t within two consecutive times exceeds a given threshold. In more detail, set the two tolerances $\texttt{tol}_s$ and $\texttt{tol}_t$, we update the spatial and/or the temporal grid if

$$\Delta\eta_s = |\eta_s(t^n) - \eta_s(t^{n-1})| > \texttt{tol}_s \quad \text{and/or} \quad \Delta\eta_t = |\eta_t(t^n) - \eta_t(t^{n-1})| > \texttt{tol}_t. \tag{25}$$

If the adaptation in space is invoked, until the first criterion in (25) is met, we go through the procedure in Sect. 3.1 to assemble $M^{n,*}$ and to generate the adapted mesh $\mathcal{T}_h^{n,*}$. We observe that a projection of the variables onto the current mesh is required during the mesh adaption loop. The outcome of such iterations coincides with the adapted mesh, identified with the grid $\mathcal{T}_h^{n+1}$, to be employed at the successive time instant.
Vice versa, if $\Delta\eta_s \leq \texttt{tol}_s$, we set $\mathcal{T}_h^{n+1} = \mathcal{T}_h^n$.
Simultaneously with the adaptation in space, the computation of the new time-step is carried out in a separate loop following the strategy in Sect. 3.2. The first time step that infringes the second condition in (25) provides the length Δt^{n+1}.

As a general remark, the space-time adaptation is constrained by setting maximum and minimum values for the mesh size and the time-step. This expedient allows avoiding over-coarsening/refinement in order to strike a balance between accuracy and efficiency.

4 Numerical assessment

To validate the approach proposed in Section 3, we exploit two well-known benchmark configurations, namely the lid-driven cavity flow [53] and the backward-facing step [54], by investigating several regimes characterized by different Reynolds numbers.

The discretization in (10) together with the space-time adaptation in Fig. 8 are implemented using the software FreeFEM [55], which offers a reliable and efficient metric-based mesh generator module.

Finally, concerning the driving quantities U_h^n and U_h^V in Sections 3.1-3.2, we opt for velocity-related information, since the strong irregularities exhibited by the pressure may jeopardize the adaptation procedure.

4.1 The lid-driven cavity flow

The lid-driven cavity flow problem has been extensively analyzed in both 2D and 3D scenarios [56], since providing a physical configuration recurrent in several engineering

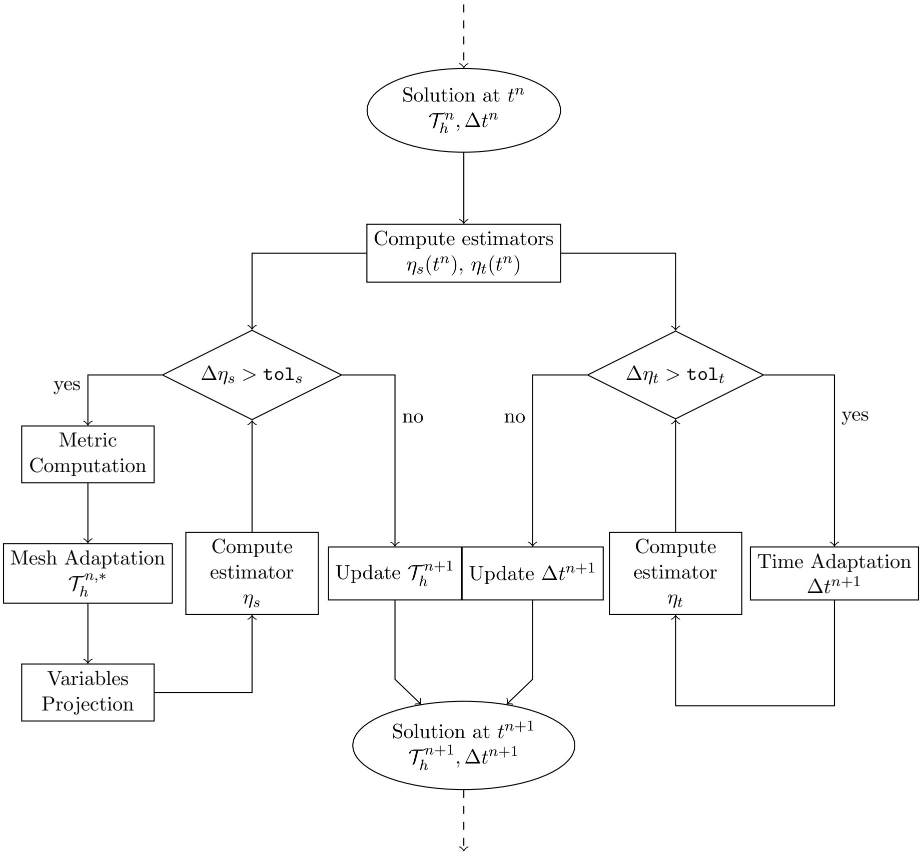


Figure 1: Sketch of the solution-adaptation algorithm

applications, such as in the design of industrial mixing processes and for the study of natural convection in enclosed spaces. Among the different settings available in the literature, we refer to the one in [53] characterized by the following data: $\Omega = (0,1)^2$; $\Gamma_D = \partial\Omega$; $\Gamma_N = \emptyset$; $t^f = 50$; $\nu = \mathbb{R}\mathrm{e}^{-1}$; $\rho = 1$; $\mathbf{f} = \mathbf{0}$; $\mathbf{u}_0 = \mathbf{0}$; $\boldsymbol{\psi} = \mathbf{0}$ and

$$\boldsymbol{\varphi} = \begin{cases} [1,0]^T & y = 1 \\ \mathbf{0} & \text{elsewhere.} \end{cases}$$

For the first set of numerical experiments, we set the Reynolds number $\mathbb{R}\mathrm{e}$ to 1000. In this flow regime, the selected final time allows us to reach the stationary state.
Concerning the discretization setting, we pick a reference mesh $\mathcal{T}_h^0$ consisting of 23876 elements uniformly distributed across Ω and an initial time-step set to $\Delta t^1 = 1\mathrm{e}{-2}$, in order to match the CFL condition. The discrete solution associated with such a discretization will provide the reference setting for the analysis below (see Fig. 2).

The framework described above is now employed to investigate the performance of the adaptive procedures in Sections 3.1-3.3, in terms of solution accuracy and computational efficiency.

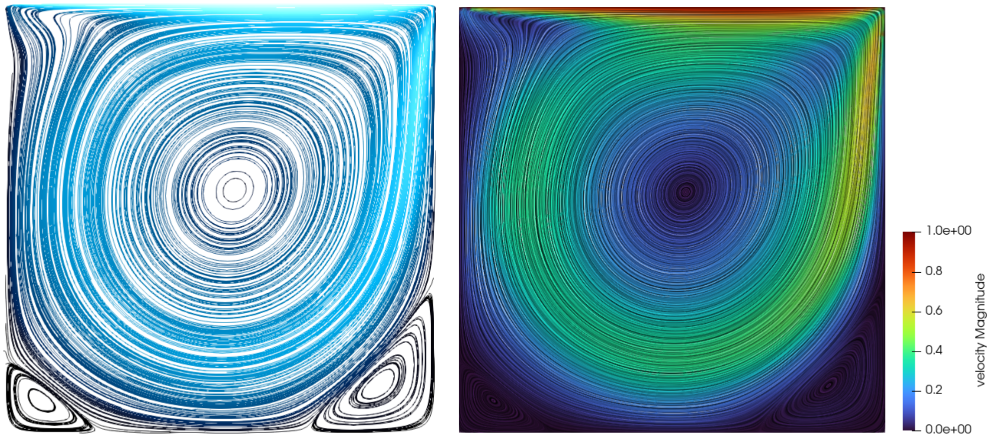


Figure 2: Lid-drive cavity flow - reference solution: streamline (left), velocity field (right).

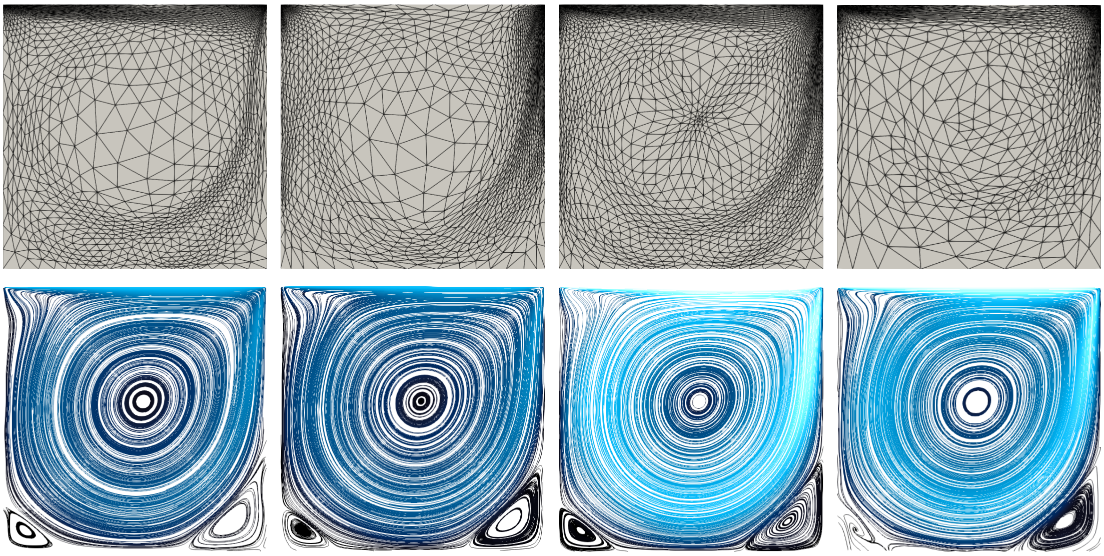
4.1.1 Adaptation in space

As a first task, we investigate the effectiveness of the spatial adaptive procedure driven by the error estimator in (14) for different choices of the driving quantity U_h^n. The mesh adaptation is initiated starting from grid $\mathcal{T}_h^0$ and by setting tolerance $\tau_s = 5\text{e}{-3}$ and $\hat{C}_K = 1$ in (19). Additionally, in order to avoid over-refinement near the top corners, we enforce a minimum value for the mesh size $h_{\min} = 1\text{e}{-5}$.
We check whether a single component of the velocity is sufficiently informative to accurately solve the whole flow dynamics in the cavity, by choosing $U_h^n = u_{h,x}^n$ and $U_h^n = u_{h,y}^n$. Columns (a)-(b) in Fig. 3 show the adapted mesh associated with the two velocity components (top) together with the associated flow streamlines (bottom) at the final time t^f. It is evident that each adapted grid captures the flow features along the tracked direction only, thus approximating the flow dynamics with a low accuracy (compare with Fig. 2, left, where the same colormap is employed).

To overcome the inaccurate detection of the flow characteristics, we combine the information associated with the two velocity components according to a mono- and a multi-criterion strategy. In the former case, we set U_h^n to the modulus $|\mathbf{u}_h^n|$ of the velocity at time t^n in (14), while the multi-criterion approach merges the metrics associated with the two driving quantities $u_{h,x}^n$ and $u_{h,y}^n$. Following [57], we are led to replace matrix $G_K(\mathcal{R}_\nabla(U_h^n) - \nabla U_h^n)$ with

$$G_K(\mathcal{R}_\nabla(u_{h,x}^n) - \nabla u_{h,x}^n) + G_K(\mathcal{R}_\nabla(u_{h,y}^n) - \nabla u_{h,y}^n).$$

Column (d) in Fig. 3 highlights that the adapted mesh yielded by the multi-criterion strategy poorly describes the main dynamics in the central portion of Ω and that the bottom-left vortex is not detected by the streamlines. On the contrary, the mono-criterion strategy guarantees an accurate modeling of the flow dynamics (compare with



(a) $U_h^n = u_{h,x}^n$ $\#\mathcal{T}_h^n = 7685$ (b) $U_h^n = u_{h,y}^n$ $\#\mathcal{T}_h^n = 5359$ (c) $U_h^n = |\mathbf{u}_h^n|$ $\#\mathcal{T}_h^n = 8660$ (d) $U_h^n = u_{h,x}^n$ & $u_{h,y}^n$ $\#\mathcal{T}_h^n = 8262$

Figure 3: Lid-driven cavity flow – adaptation in space: anisotropic adapted mesh (top) and associated flow streamlines (bottom) at the final time t^f for different choices of the driving quantity U_h^n.

Fig. 2, left), with a reduction in computational time from 501 to 280 minutes[1] (i.e., 23876 versus 8660 elements). The efficiency of the space adaptation algorithm is further confirmed by cross-comparing the performance of the anisotropic algorithm with the corresponding isotropic implementation, which demands 669 minutes (i.e., 19872 triangles) to ensure the same accuracy $\tau_s = 5\text{e}{-3}$ (see Fig. 4).

To sum up, we will adopt $U_h^n = |\mathbf{u}_h^n|$ as the driving quantity for the space adaptation, since providing the best performance so far.

4.1.2 Adaptation in time

Regarding the time adaptive procedure, we adopt the same driver used to update the tessellation $\mathcal{T}_h$, namely we pick $U_h^V = |\mathbf{u}_h|$ in (20). We set the tolerance τ_t in (23) to 8e−1 with $\Delta t^1 = 1\text{e}{-2}$, and we adopt the fixed mesh $\mathcal{T}_h^0$ in Fig. 2 as spatial discretization. Moreover, the upper bound $\Delta t_{\max} = 1.5$ is imposed to avoid over-coarsening when the flow reaches the stationary state.

In Fig. 5 (left), we display the evolution of Δt in time, which gradually increases until reaching the prescribed limit $\Delta t_{\max}$ when the flow stabilizes. Notice that, there is a considerable discrepancy in terms of order of magnitude between the fixed time-step Δt^1 in Fig. 2 and the average value (i.e., $\{\Delta t\} = 0.9$) of the predicted adapted time-step

---

[1] The computations have been run on a 12th Gen Intel® Core™ i7-1260P, Mesa Intel® Graphics, 16 GB RAM desktop computer.

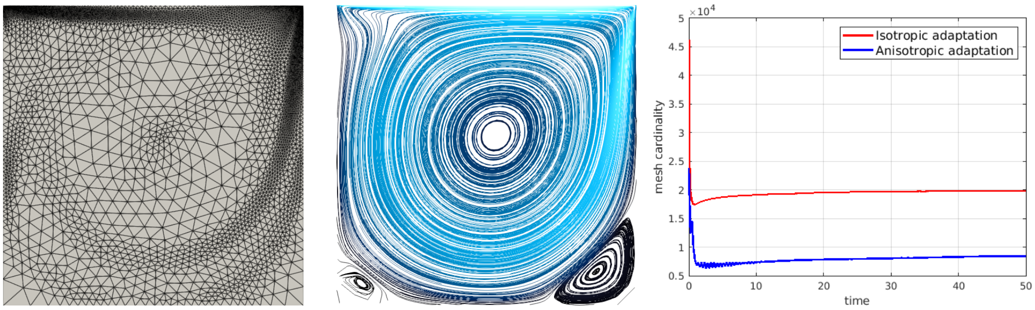


Figure 4: Lid-driven cavity flow – adaptation in space: isotropic adapted mesh (left) and associated flow streamlines (middle) at the final time t^f; comparison between the isotropic and anisotropic mesh adaptation in terms of time evolution of the number of triangles (right).

in Fig. 5, left. Unavoidably, this has an impact onto the computational time that is reduced from 501 to 27 minutes.

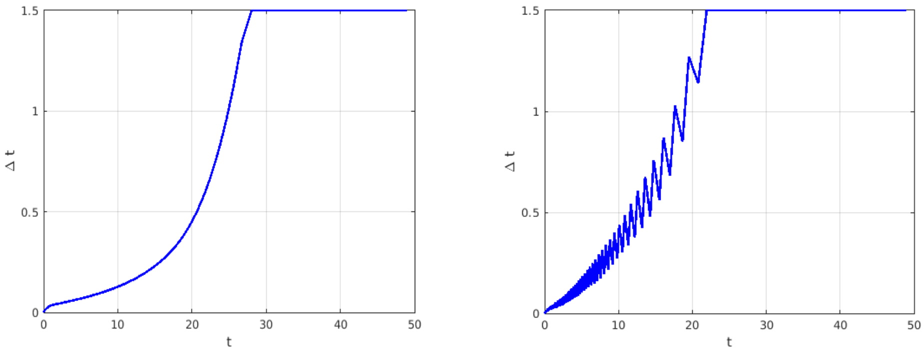


Figure 5: Lid-driven cavity flow – adaptation in time: evolution of the time step for a fixed (left) and an adapted (right) spatial mesh.

To corroborate the reliability of the time adaptation procedure, we compare the obtained velocity profiles with results consolidated in the literature, referring, in particular, to [53]. The maximum percentage error associated with components $u_{h,x}$ and $u_{h,y}$ of the flow velocity is 0.08% and 0.13%, respectively, in terms of the maximum norm.
As a last check, motivated by the notable difference between Δt^1 and $\{\Delta t\}$ and to validate the usefulness of the adaptation procedure in time, we approximate the flow velocity on $\mathcal{T}_h^0$ after setting $\Delta t^1 = \{\Delta t\} = 0.9$. This assessment highlights that the maximum percentage error increases to 4.18% and 3.69% for $u_{h,x}$ and $u_{h,y}$, respectively when resorting to a fixed space-time discretization. The benefits yielded by the procedure in Sect. 3.2 are thus confirmed.

4.1.3 Space-time adaptation procedure

This section emphasizes the advantages ensured by the combination of anisotropic mesh with time-step adaptation, following the strategy in Sect. 3.3. In particular, in (25) we assign the values 1e−6 and 1e−4 to the tolerances $\texttt{tol}_s$ and $\texttt{tol}_t$, while preserving $\tau_s = 5\text{e}{-3}$ and $\tau_t = 8\text{e}{-1}$ to constrain the adaptation in space and time, respectively. The initial mesh $\mathcal{T}_h^0$ and time-step Δt^1 do coincide with the ones employed in Sections 4.1.1-4.1.2.

The anisotropically adapted mesh together with the associated flow streamlines at the final time t^f are provided in Fig. 6 (a)-(b) – exhibiting a good match with the reference solution in Fig. 2, left – while the evolution of the time step, Δt^n, is provided in Fig. 5, right. The cardinality of the final adapted mesh in Fig. 3 (c) and in Fig. 6 (a) is comparable (8660 vs 8370), as well as the distribution of the streamlines in Ω with a reliable description of the main vortices in both cases. Concerning the adaptation in time, a cross-comparison between the two plots in Fig. 5 showcases a common trend towards the upper bound for Δt, with a higher incremental rate for the space-time adaptation, leading to 222 instants instead of 270 times required by the time adaptation alone. Moreover, the irregular trend in the right panel can be ascribed to the adaptation in space and to the corresponding variable projection onto the new tessellation.

As expected, the space-time adaptation procedure yields the lowest computational effort in terms of CPU time when compared with a fixed space-time discretization, a partial (in space or in time only) adaptation, both in an isotropic and an anisotropic context (see Table 1). In particular, the space-time process in Fig. 1 requires 15 minutes to be delivered, corresponding to a 97.2% computational saving with respect to the time (501 minutes) characterizing the simulation in Fig. 2. The combination of time with space adaptation halves the requested CPU time when compared with an adaptation in time only (running in 27 minutes), and features a 94.64% and 97.75% CPU time reduction when compared with an anisotropic and isotropic adaptation in space only (running in 280 and 669 minutes), respectively.

The total computational time of the space-time adaptive procedure is unequally distributed among the various tasks involved in the procedure, as emphasized in Table 2. About half of the whole time is required to carry out the adaptation in space, whereas a limited percentage ($\sim$ 3%) of the CPU time is employed by the time adaptation. This time discrepancy can be partly explained by the difference in terms of dimensionality between the spatial and the temporal domains (2D vs 1D, respectively).

	$\#\mathcal{T}_h^f$	$\#\Delta t$	CPU time
Fixed space-time discretization	23876	5000	501 min
Anisotropic space adaptation	8660	5000	280 min
Isotropic space adaptation	19872	5000	669 min
Time adaptation	23876	270	27 min
Anisotropic space + time adaptation	8370	222	15 min

Table 1: Lid-driven cavity flow: comparison among different space-time discretizations.

Task	CPU time
System assembly and solver	338.329 s
Spatial error estimation	285.233 s
Metric computation	51.727 s
Anisotropic mesh adaptation	63.942 s
Time error estimation	27.667 s
Time adaptation	0.00186 s

Table 2: Lid-driven cavity flow - adaptation in space and time: CPU time associated with the different tasks.

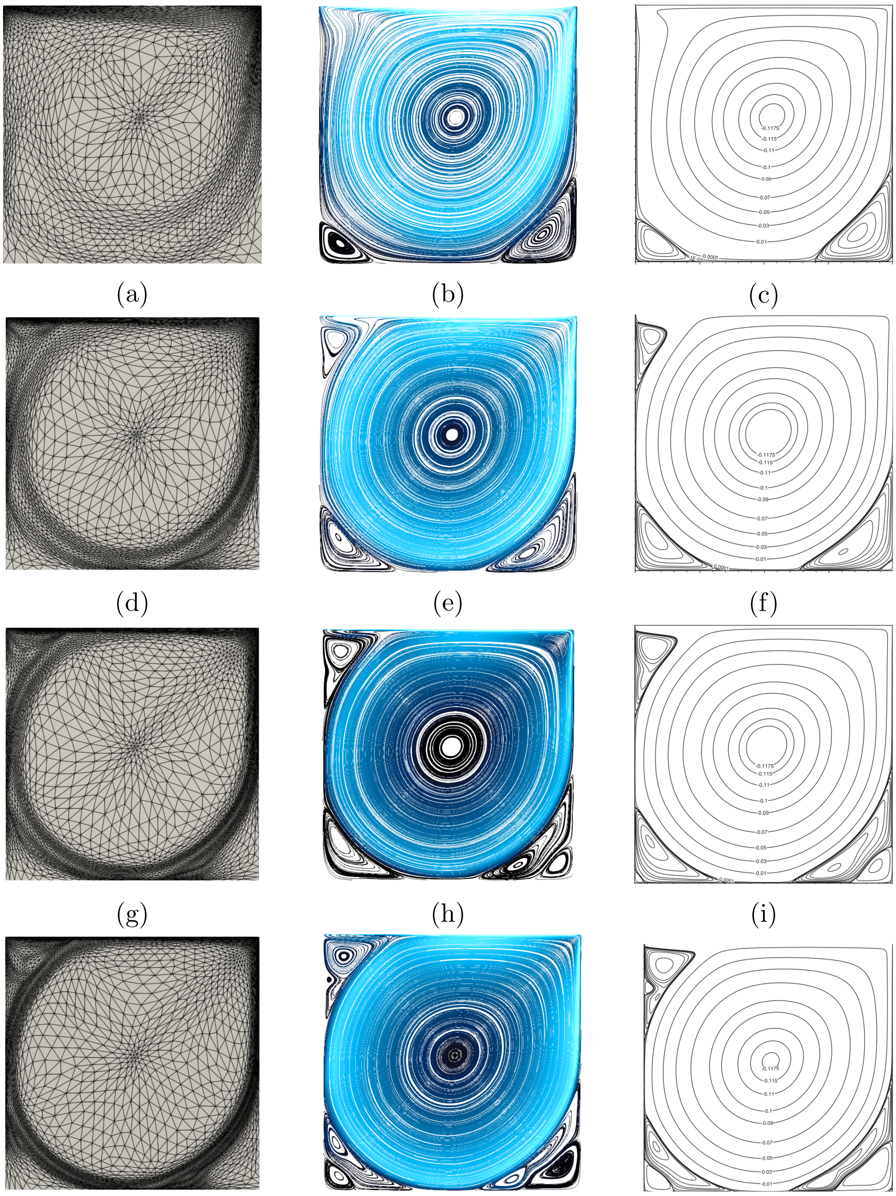
To challenge the robustness of the space-time adaptive procedure, we choose higher values for the Reynolds number towards the modeling of an actual turbulent flow. As interval of variation, we consider values ranging from $\mathbb{R}\mathrm{e} = 1000$ – i.e., a completely laminar flow – to $\mathbb{R}\mathrm{e} = 20000$ – exceeding the experimentally-determined critical value $\mathbb{R}\mathrm{e}_{\mathrm{cr}} = 8000$ when instability phenomena begin to emerge. In particular, when dealing with high values for $\mathbb{R}\mathrm{e}$, we have to extend the final time in order to ensure that the flow fully develops and stabilizes. Additionally, it is necessary to use a finer space and time discretization at t^0 and to properly tune the space-time mesh adaptation in order to accurately capture the whole flow dynamics development. Table 3, left panel, collects the exact data adopted for the different choices of $\mathbb{R}\mathrm{e}$. Figure 6 gathers the adapted mesh at the final time (on the left), the associated streamlines (in the middle). In particular, we are interested in verifying the number and the position of the vortices. To this aim, we adopt the results in [53] as reference (right column in Fig. 6).

Despite the increase in the number of vortices and in the characterizing complexity when $\mathbb{R}\mathrm{e}$ becomes larger and larger, the approximation of the flow turns out to be very accurate when compared to the reference solution. This result is obtained thanks to increasingly finer adapted meshes characterized by high aspect ratios, and with longer computational times that, however, remain sustainable (see the right panel in Table 3).

Input data						Output data				
$\mathbb{R}\mathrm{e}$	t_f	τ_s	τ_t	$\#\mathcal{T}_h^0$	Δt^1	$\#\mathcal{T}_h^f$	$\max_K s_K$	$\{\Delta t^n\}$	$\#\Delta t$	CPU time
1000	50	5e−3	8e−1	23876	1e−2	8454	90.41	0.2272	222	15 min
5000	150	5e−3	9e−1	34186	5e−3	10097	106.58	0.3008	501	42 min
10000	400	5e−3	1	60426	1e−3	11270	179.50	0.4915	816	70 min
20000	800	5e−3	1	94866	1e−3	12883	188.51	0.4888	1639	170 min

Table 3: Lid-driven cavity flow - adaptation in space and time: some of the input (left) and output (right) data.

To make the comparison with the reference solution more quantitative, in Fig. 7 we superimpose the velocity profiles yielded by space-time adaptive approach to the results in [53], tracked for $u_{h,x}$ and $u_{h,y}$ along the straight lines $y = 0.5$ and $x = 0.5$, respectively. The output of the proposed procedure (solid blue line) is coherent with the values in [53] (red markers). This holds for both the velocity components, independently of the Reynolds number value, and also in the presence of steep gradients.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6: Lid-driven cavity flow - adaptation in space and time: sensitivity analysis to $\mathbb{R}\mathrm{e}$ ($= 1000, 5000, 10000, 20000$, top-bottom) in terms of anisotropically adapted mesh at t^f (left), number and location of vortices (middle) with respect to the results in [53] (right).

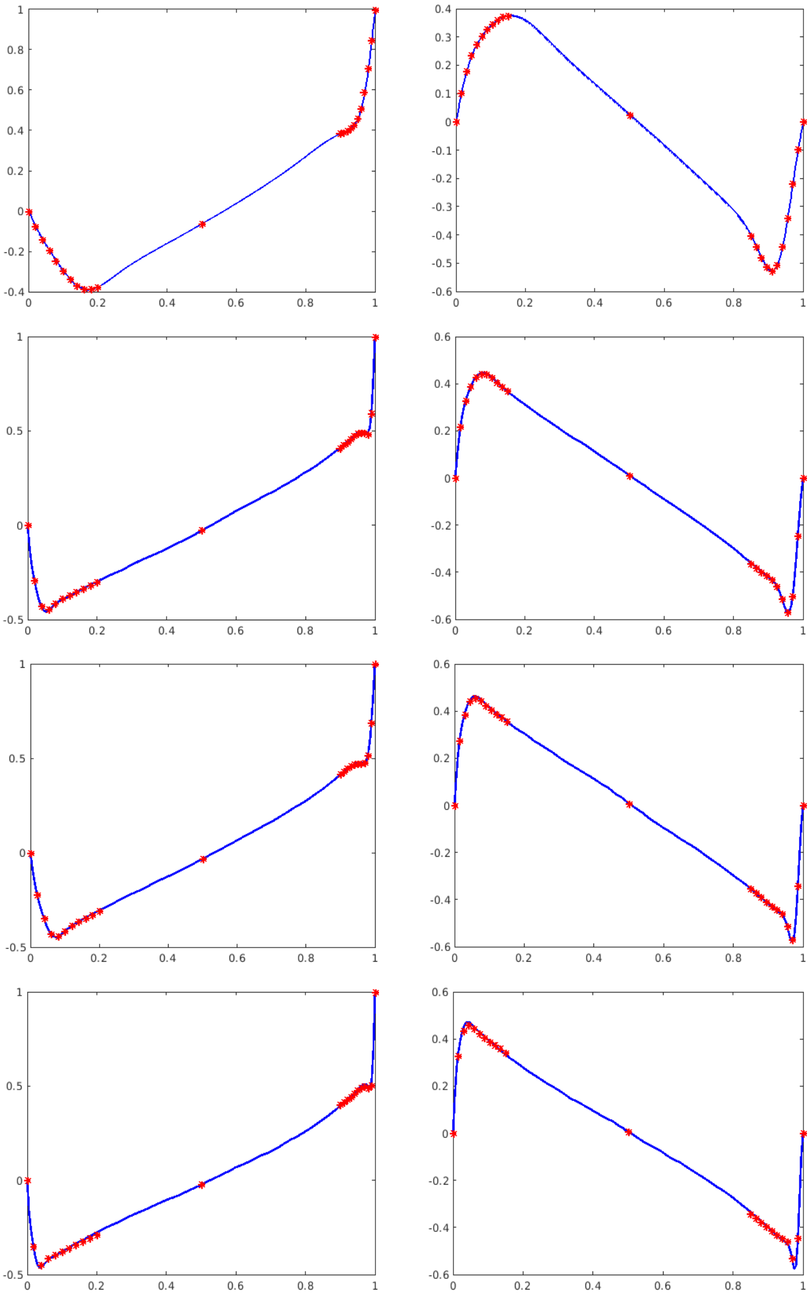


Figure 7: Lid-driven cavity flow - adaptation in space and time: comparison between the output of the procedure with the results in [53] at the final time, in terms of the two components, $u_{h,x}$, (left) and, $u_{h,y}$ (right) along lines $y = 0.5$ and $x = 0.5$, respectively, for $\mathbb{R}\mathrm{e} = 1000, 5000, 10000, 20000$ (top-bottom).

Finally, Fig. 8 corroborates the reliability and efficiency of the space-time adaptation when dealing with the most challenging fluid dynamics scenario here considered ($\mathbb{R}\mathrm{e} =$

20000).

The spatial adapted mesh, combined with the adaptive selection of the time-step, allows us to exactly follow the turbulent flow evolution throughout the whole time window with an affordable computational time. Conversely, performing the same simulation with a fixed space-time discretization that is sufficiently fine to capture the flow details (e.g., $\#\mathcal{T}_h^0 = 94866$, $\Delta t^1 = 3\text{e}{-3}$) proves to be computationally prohibitive when using the same resources.

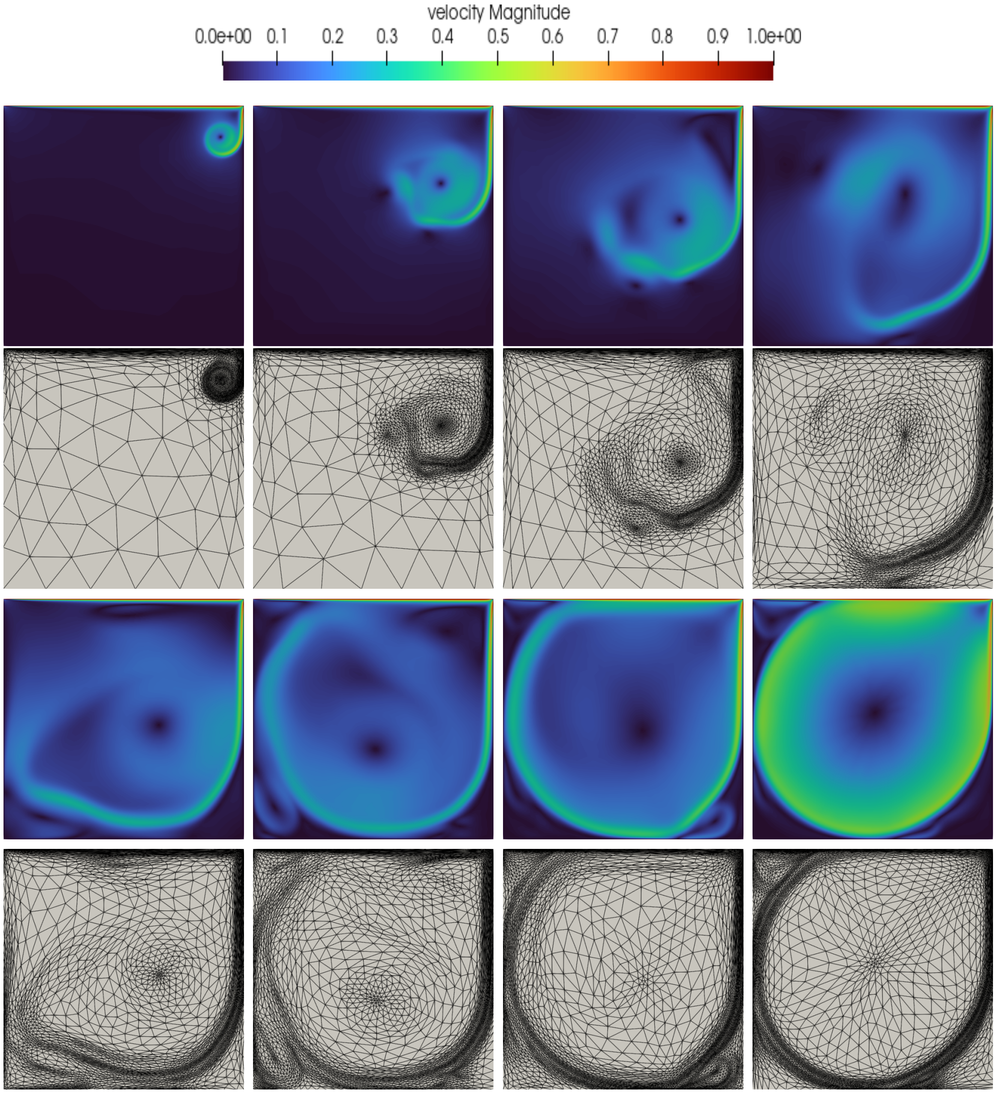


Figure 8: Lid-driven cavity flow - adaptation in space and time at $\mathbb{R}\text{e} = 20000$: time evolution of the flow velocity magnitude and the underlying adapted computational mesh for $t = 2.0052, 7.00736, 12.0304, 17.1572, 22.0722, 27.1545, 42.3262, 800$.

With reference to the specific choice made for the space-time adaptation algorithm, we observe that spatial adaptation only slightly changes the mesh cardinality (see Fig. 9, left). On the contrary, temporal adaptation leads to more significant variations in the time step in order to accommodate the complex flow dynamics (see Fig. 9, right) and due to the projection step in the scheme of Fig. 1, in accordance with what observed in Fig. 5, right. The overall oscillatory behavior of both plots in Fig. 9 is due to the difficulty in reaching the flow's stationary state.

The values in Table 1 and the results Figs. 6–8 support the selection of the space-time adaptation, with an anisotropic tessellation of Ω, as the most strategic approach for the modeling of flows, independently of the selected Reynolds number.

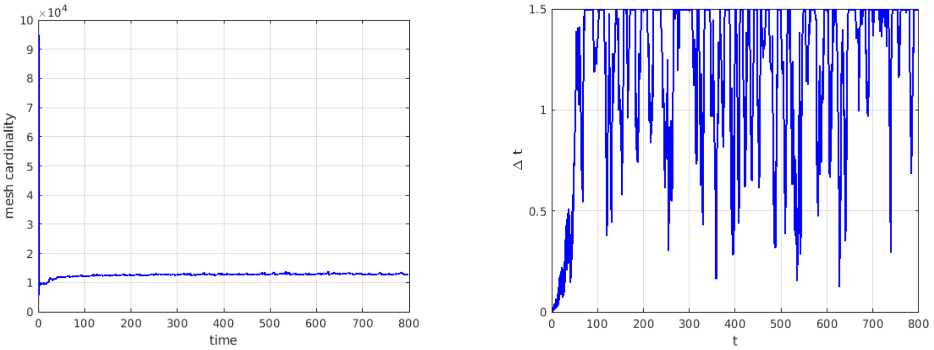


Figure 9: Lid-driven cavity flow - adaptation in space and time at $\mathbb{R}\mathrm{e} = 20000$: evolution of the mesh cardinality (left) and of the time step (right).

4.2 The backward-facing step

The second test case we consider is the backward-facing step (BFS), a classic fluid dynamics configuration frequently used in engineering in order to study separation and reattachment phenomena. Typically, the BFS geometry involves a two-dimensional duct featuring a step, leading to an abrupt increase in the cross-sectional area. As a result, the flow separates from the surface, forming a recirculation zone past the step - a region characterized by high turbulence and mixing - before eventually reattaching further downstream. The step height, the variation of the cross-sectional area, and the Reynolds number are critical parameters that influence the flow characteristics.

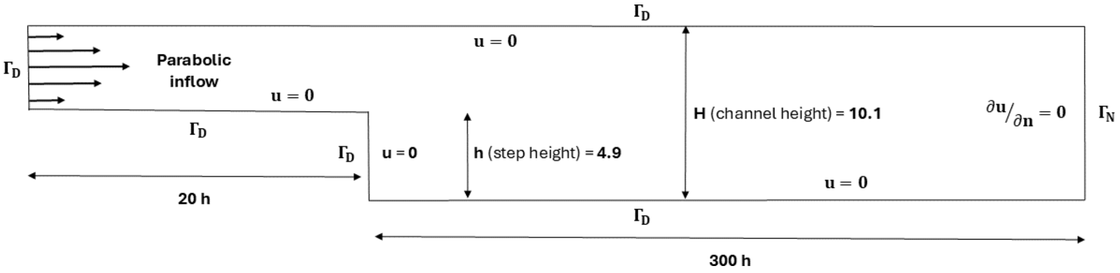


Figure 10: Backward facing step geometry

The reference setting we adopt is based on the work in [54]. Specifically, the domain Ω and the boundary portions Γ_D and Γ_N are illustrated in Fig. 10. The problem data are set as follows: $\nu = 1\text{e}{-1}$; $\rho = 1$; $\mathbf{f} = \mathbf{0}$; $\psi = \mathbf{0}$; $\mathbf{u}_0 = \mathbf{0}$ with the inlet velocity $\boldsymbol{\varphi}$ defined as

$$\boldsymbol{\varphi} = \begin{cases} \left[\dfrac{3\nu\mathbb{R}\text{e}}{(H-h)}\,(H-y)\,(y-h)\,,0\right]^T & x = 0 \\ \mathbf{0} & \text{elsewhere,} \end{cases}$$

where $\mathbb{R}$e ranges from 100 to 2000. The channel height is $H = 10.1$, and the step height is $h = 4.9$, corresponding to an expansion ratio, $H/(H-h)$, equal to 1.942. Concerning the final time, t_f, we have selected the time window according to the flow regime in order to limit the whole computational budget of the simulation (see Table 4, left panel).
For the BFS setting, it is well-established that the computational mesh has to be fine enough in order to sharply approximate the flow behavior near walls. As an alternative, wall laws, such as the well-known logarithmic function [58, 59, 60], are used [61, 62, 63] as modeling correction, in combination with moderately fine grids (see, e.g., [64, 65, 66, 67] for an example in the VMS framework). However, using a fixed uniform mesh with an element size sufficiently small or the wall law correction on a medium-resolution grid leads, in general, to a prohibitive computational effort.
In this section, we pursue a different approach by skipping the wall law model and relying on an adapted mesh, which is selectively refined where high velocity gradients occur. In particular, we resort to the space-time adaptive scheme in Fig. 1, motivated by the computational benefits yielded by the adaptive time-step selection when combined with the spatial adaptation (see Table 3). Instances of mesh adaptation applied to the BFS test case are available in the literature. In [21], in a 2D context, the VMS-Smagorinsky model is enriched with an isotropic mesh adaptation for a fixed time-step. In [18, 22], the authors resort to an anisotropic adapted mesh to model the flow dynamics for the 3D BFS. In particular, in [18], no turbulence correction is included in the CFD model and the time-step is tuned by a heuristic CFL-based adaptive criterion. In [22], an anisotropic error indicator based on the energy associated with the resolved small scales drives the grid adaptation and the time-step is kept fixed.

We initiate the space-time adaptive procedure using a uniform initial mesh, $\mathcal{T}_h^0$, consisting of 32422 elements. The initial time-step, Δt^1, is determined in order to match the CFL condition according to the selected initial velocity magnitude (i.e., the considered Reynolds number). We choose the velocity magnitude as driving quantity for the whole adaptation process, and we set the associated parameters as $\hat{C}_K = 1$, $\Delta t_{max} = 1.5$, $h_{\min} = 1\text{e}{-1}$, $\texttt{tol}_\texttt{s} = 1\text{e}{-6}$ and $\texttt{tol}_\texttt{t} = 1\text{e}{-4}$, while tolerances τ_s and τ_t are tuned according to the flow regime.

We consider five scenarios characterized by increasingly complex fluid dynamics, specifically with progressively higher values of $\mathbb{R}$e, ranging from a laminar ($\mathbb{R}\text{e} = 100$, 500) to a turbulent ($\mathbb{R}\text{e} = 1000$, 1500, 2000) regime. As the Reynolds number increases, maintaining stability becomes challenging due to the formation of steep velocity gradients close to the wall. To address this issue, we reduce the upper bound of the CFL condition from 1 to 0.8. Additionally, we limit abrupt changes between two consecutive

anisotropic adapted meshes, $\mathcal{T}_h^{n-1}$ and $\mathcal{T}_h^n$, by employing a metric relaxation algorithm. Figures 11–15 gather the distribution of the flow velocity magnitude (top panel), the adapted mesh at the final time (center panel), and the streamlines (bottom panel) for the different choice of the Reynolds number. In particular, figures are related to a specific portion of Ω where the more involved dynamics take place. This is instrumental to identify the separation areas as well as the reattachment point coordinates.
As expected, when the value of $\mathbb{R}\mathrm{e}$ increases, the fluid dynamics becomes more complex, featuring a higher velocity (compare the different scales) and a larger number of vortices. Accordingly, anisotropic adapted meshes are progressively refined, allowing for an accurate track of the fluid separation and of the reattachment points, which are correctly identified in accordance with [54]. The right panel in Table 4 offers quantitative information about the space-time adaptive simulations. When moving from a laminar to a turbulent regime, we observe that the average time-step, $\{\Delta t^n\}$, diminishes (i.e., $\#\Delta t$ increases) and the first check in (25) demands more iterations to be satisfied. These remarks justify the increment in terms of CPU time, which still remains affordable, as confirmed by the values in the table.

As a further check about the adequacy of the spatial adapted grid, we rely on the quantity

$$y^+(\mathbf{x}) = \frac{u_\tau(\mathbf{x}) h_w(\mathbf{x})}{\nu}$$

introduced in [68], where u_τ is the friction velocity defined as $u_\tau = \sqrt{\frac{\tau_w}{\rho}}$, where τ_w is the wall shear stress and ρ is the fluid density, and h_w denotes the distance from the wall. Quantity y^+ measures the relative importance of viscous and turbulent processes, similarly to the local Reynolds number. To verify that the mesh is correctly refined close to the wall in order to ensure an accurate modeling of the boundary layer, as a practical rule, we consider the mesh to be adequate when $y^+ < 5$ close to the boundary. Vice versa, wall law corrections are deserved [68, 64].

Concerning the results in Figs. 11-15, the value of y^+ computed according to the equations in [64] is close to zero. This corroborates mesh adaptation strategy as a reliable alternative choice to a uniform mesh refinement as well as to wall law corrections. In addition, the proposed adaptive approach offers substantial computational saving compared to using a wall correction model. For instance, for $\mathbb{R}\mathrm{e} = 2000$, ensuring $y^+ < 5$ with the wall function in [64] results in a prohibitive cost of 14 seconds on average for each instant. In contrast, this time reduces to just 4.12 seconds when resorting to the proposed adaptive method.

Input data					Output data				
$\mathbb{R}\mathrm{e}$	t_f	τ_s	τ_t	Δt^1	$\#\mathcal{T}_h^f$	$\max_K s_K$	$\{\Delta t^n\}$	$\#\Delta t$	CPU time
100	1043	4e−2	2	0.7189	2224	30.85	1.4694	702	25 min
500	933	8e−2	4	0.1438	3508	56.04	0.7483	955	66 min
1000	832	2e−1	16	0.0647	3340	51.41	0.4716	1764	89 min
1500	733	2e−1	256	0.0431	4506	71.14	0.3962	1749	113 min
2000	632	3e−1	1000	0.0324	4325	59.21	0.2933	2127	146 min

Table 4: Backward facing step - adaptation in space and time: some of the input (left) and output (right) data.

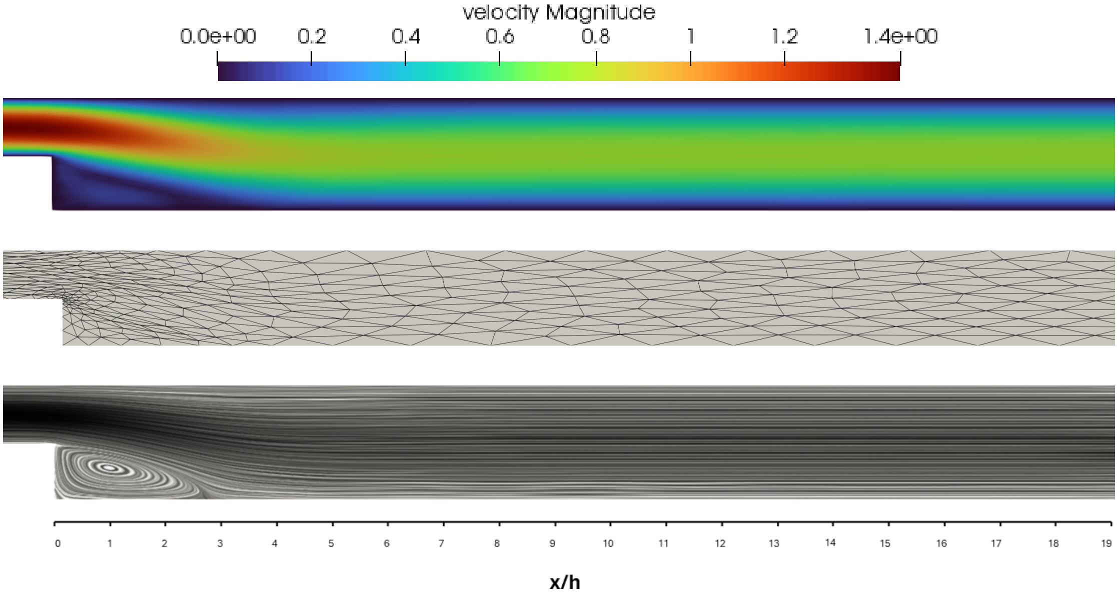


Figure 11: Backward-facing step – adaptation in space and time at $\mathbb{R}\mathrm{e} = 100$: flow velocity magnitude (top), adapted computational mesh (middle), and associated streamlines (bottom) at the final time t^f.

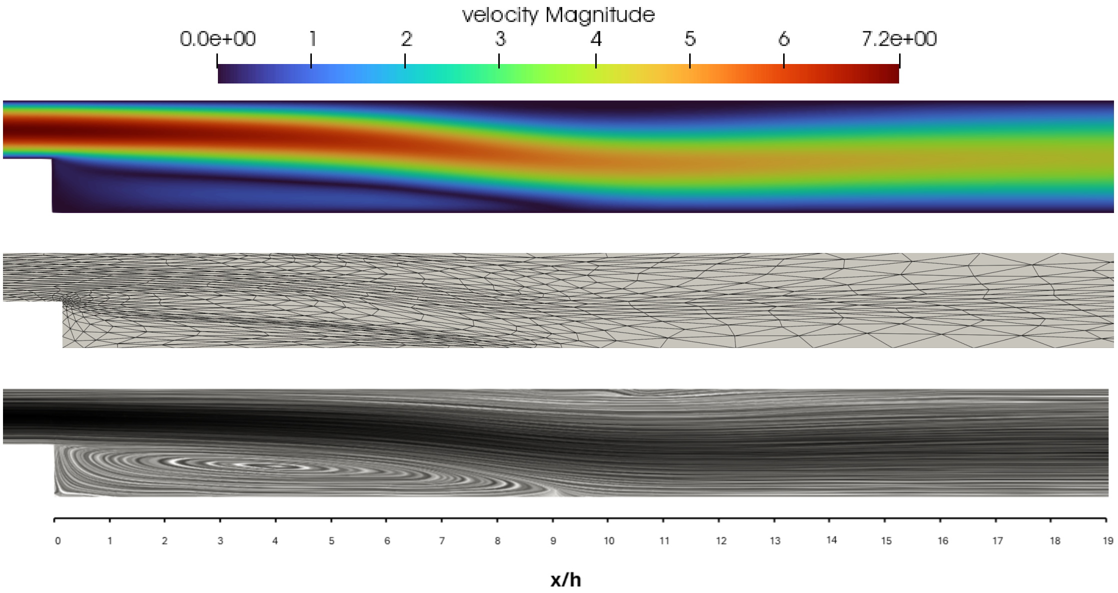


Figure 12: Backward-facing step – adaptation in space and time at $\mathbb{R}\mathrm{e} = 500$: flow velocity magnitude (top), adapted computational mesh (middle), and associated streamlines (bottom) at the final time t^f.

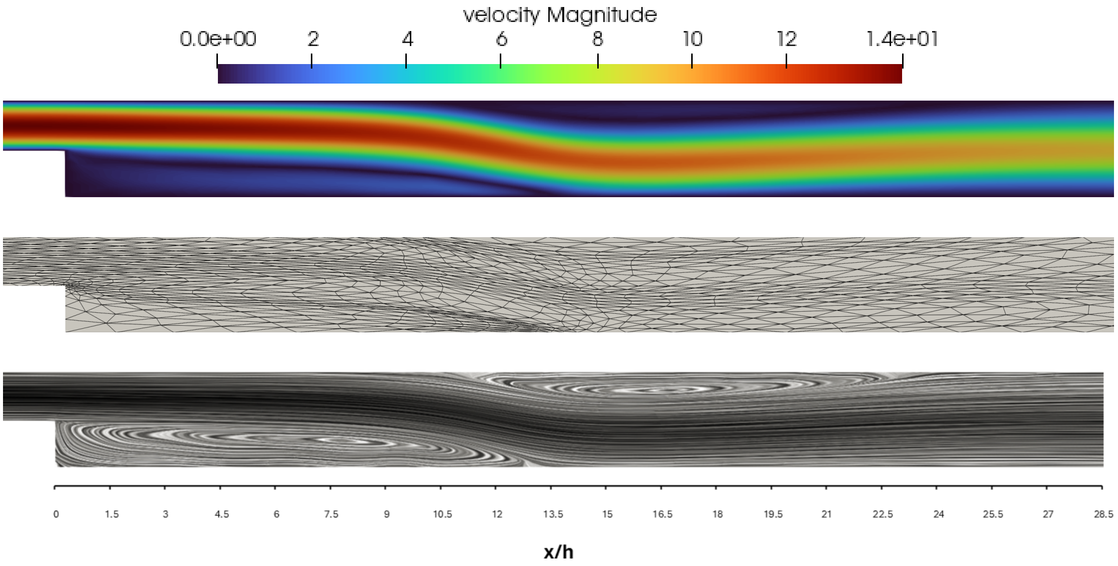


Figure 13: Backward-facing step – adaptation in space and time at $\mathbb{R}\mathrm{e} = 1000$: flow velocity magnitude (top), adapted computational mesh (middle), and associated streamlines (bottom) at the final time t^f.

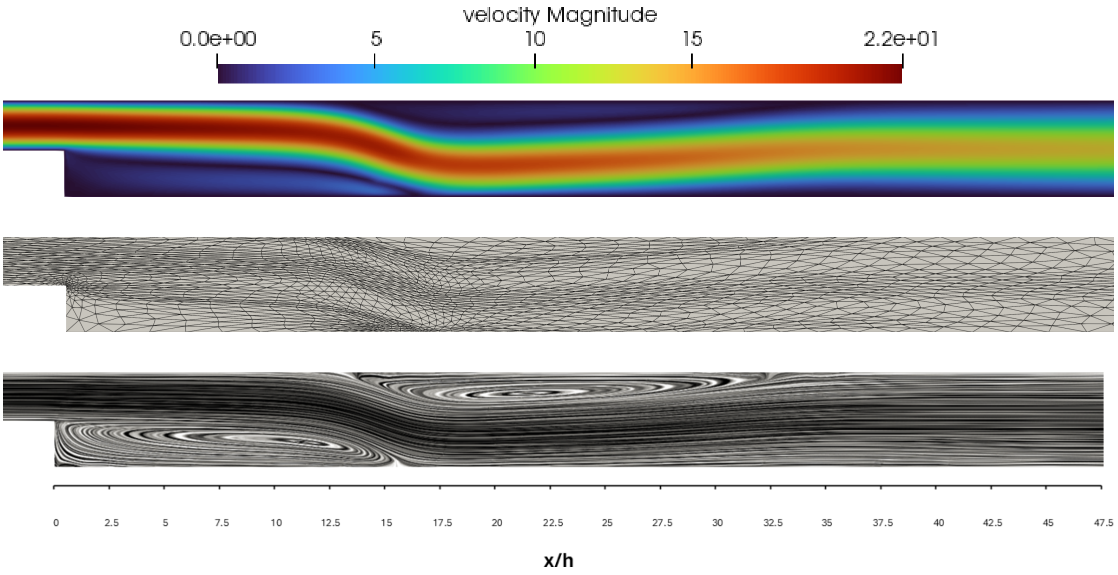


Figure 14: Backward-facing step – adaptation in space and time at $\mathbb{R}e = 1500$: flow velocity magnitude (top), adapted computational mesh (middle), and associated streamlines (bottom) at the final time t^f.

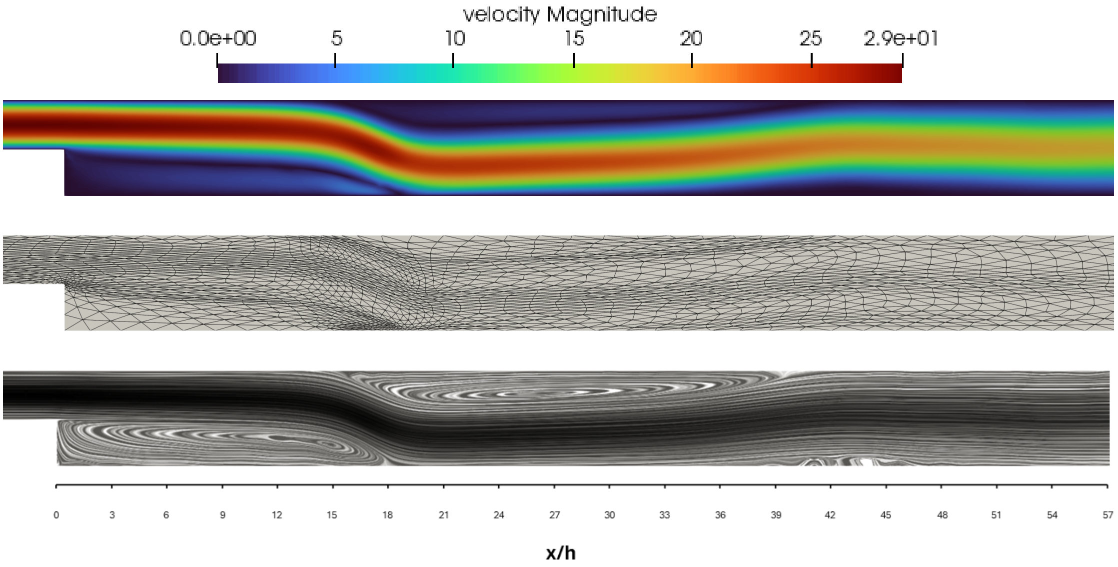


Figure 15: Backward-facing step – adaptation in space and time at $\mathbb{R}e = 2000$: flow velocity magnitude (top), adapted computational mesh (middle), and associated streamlines (bottom) at the final time t^f.

5 Conclusions

In this paper we focus on benefits and limits yielded by a space-time adaptive procedure tailored to 2D high Reynolds number flows cast in a VMS-Smagorinsky turbulence model. The computational mesh and the time window partition are selected through an anisotropic and an isotropic a posteriori recovery-based error estimator, respectively. As a first task, we have identified the velocity modulus as a reliable quantity to drive the adaptation algorithm. Concerning the spatial discretization, anisotropic adapted meshes confirm the capability in sizing, shaping and orienting triangles in order to follow the intrinsic directionality of the flow, lowering the required CPU time when compared to a fixed mesh setting or to an isotropic adaptive strategy. In particular, anisotropic grids are capable of accurately resolving singularity regions and near-wall flow behavior, without excessively increasing the number of mesh elements and avoiding any wall law correction. As far as the time discretization is concerned, the implemented time-step adaptive strategy allows for computational savings, especially when the flow reaches a stationary condition, while ensuring small time-steps during phases of rapid flow change to guarantee reliable accuracy. Results in Table 1 for the lid-driven cavity flow corroborate the computational benefits led by the space and by the time adaptive algorithms, separately as well as by the combination of the two adaptive strategies. In more detail, the single space or time adaptation leads to a reduction of the CPU time by a factor equal to 18. This gain increases to 33 when resorting to a whole space-time adaptation procedure.

The reliability of the proposed adaptive discretization has been challenged on increasingly complex fluid dynamics configurations and flow regimes. In particular, the lid-driven cavity flow setting allows us to span from a laminar to an actual turbulent flow, the Reynolds number ranging from 1000 to 20000. On the contrary, the abrupt changes characterizing the BFS geometric configuration fuel strong velocity gradients near the walls, thus limiting the Reynolds number variation from 100 to 2000. Independently of the case study and of the flow regime, the space-time adaptive algorithm proved to be effective in accurately capturing flow features with an affordable computational effort, even in areas with singularities for the lid-driven cavity flow and in the ricirculation zone past the step for the BFS. As a last check, we have assessed the consistency of the output of our algorithm with respect to the literature. To this aim, we have verified the number and the position of the vortices together with the velocity profile along specific sections of the domain for the lid-driven cavity setting, and the reattachment point coordinates for the BFS. Results in Figs. 6-7 and Figs. 11-15 are fully compliant with the reference works.

The extensive numerical investigation has highlighted that the procedure in Fig. 1 requires additional care, when the flow is subject to instabilities, as for the BFS configuration when Reynolds number increases. An actual remedy consists in reducing the upper bound for the CFL condition and in limiting abrupt changes in the evolution of the anisotropic spatial mesh. In addition, the selection of tolerances τ_s and τ_t constraining the space and the time adaptation, respectively, as well as of tolerances $\texttt{tol}_s$ and

$\texttt{tol}_t$ involved in the space-time adaptive process is not currently automatized, with a consequent time consumption.
Thus, automating this process together with the extension to a 3D context constitute a mandatory step in enhancing the practicality and scalability of the proposed space-time adaptive method, with a view to broader applications.

Acknowledgments

This research was conducted within the framework of the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions, grant agreement 872442 (ARIA, Accurate Roms for Industrial Applications).
ET, NF, SP are members of the Gruppo Nazionale Calcolo Scientifico–Istituto Nazionale di Alta Matematica (GNCS–INdAM). NF acknowledges the INdAM–GNCS 2024 Project "Accuratezza geometrica e adattività di griglia per problemi con interfacce complesse".
GS was funded by the European Union (ERC, DANTE, GA-101115741). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.
Moreover, the authors thank professor Nicola Parolini for his valuable advice and feedback throughout the course of this work.

References

[1] F. R. Menter, M. Kuntz, R. Langtry, Ten years of industrial experience with the SST turbulence model, in: Turbulence, Heat and Mass Transfer, Begell House, 2003.

[2] P. R. Spalart, Strategies for turbulence modelling and simulations, Int. J. Heat Fluid Flow 21 (3) (2000) 252–263.

[3] R. Temam, Navier–Stokes equations: Theory and Numerical Analysis, Vol. 343, American Mathematical Society, 2024.

[4] P. A. Durbin, B. A. Pettersson Reif, Statistical Theory and Modeling for Turbulent Flows, John Wiley & Sons, 2011.

[5] D. C. Wilcox, Turbulence Modeling for CFD, Vol. 2, DCW Industries, 1998.

[6] P. Sagaut, Large Eddy Simulation for Incompressible Flows: an Introduction, Springer Science & Business Media, 2005.

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

[8] J. Smagorinsky, General circulation experiments with the primitive equations: I. the basic experiment, Monthly Weather Review 91 (3) (1963) 99–164.

[9] A. Beck, D. Flad, C. D. Munz, Deep neural networks for data-driven LES closure models, J. Comput. Phys. 398 (2019) 108910.

[10] R. Codina, Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, Comput. Methods Appl. Mech. Engrg. 191 (39-40) (2002) 4295–4321.

[11] T. J. R. Hughes, L. Mazzei, K. E. Jansen, Large eddy simulation and the variational multiscale method, Comput. Vis. Sci. 3 (2000) 47–59.

[12] S. Hijazi, G. Stabile, A. Mola, G. Rozza, Data-driven POD-Galerkin reduced order model for turbulent flows, J. Comput. Phys. 416 (2020) 109513.

[13] 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. Eng. 166 (1-2) (1998) 3–24.

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

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

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

[17] L. Formaggia, S. Perotto, New anisotropic a priori error estimates, Numer. Math. 89 (4) (2001) 641–667.

[18] S. Micheletti, S. Perotto, P. E. Farrell, A recovery-based error estimator for anisotropic mesh adaptation in CFD, Bol. Soc. Esp. Mat. Apl. SeMA (50) (2010) 115–137.

[19] O. Antepara, O. Lehmkuhl, R. Borrell, J. Chiva, A. Oliva, Parallel adaptive mesh refinement for large-eddy simulations of turbulent flows, Comput. Fluids 110 (2015) 48–61.

[20] P. Benard, G. Balarac, V. Moureau, C. Dobrzynski, G. Lartigue, Y. D'Angelo, Mesh adaptation for large-eddy simulations in complex geometries, Int. J Numer. Methods Fluids 81 (12) (2016) 719–740.

[21] C. Bernardi, T. Chacón Rebollo, M. Gómez Mármol, Error analysis of a subgrid eddy viscosity multi-scale discretization of the Navier-Stokes equations, SeMA Journal (60) (2012) 51–74.

[22] S. Toosi, J. Larsson, Anisotropic grid-adaptation in large eddy simulations, Comput. Fluids 156 (2017) 146–161.

[23] G. M. Porta, S. Perotto, F. Ballio, A space-time adaptation scheme for unsteady shallow water problems, Math. Comput. Simul. 82 (12) (2012) 2929–2950.

[24] B. Esfandiar, G. Porta, S. Perotto, A. Guadagnini, Impact of space-time mesh adaptation on solute transport modeling in porous media, Water Resour. Res. 51 (2) (2015) 1315–1332.

[25] A. Ern, J. L. Guermond, Theory and Practice of Finite Elements, Vol. 159, Springer, 2004.

[26] D. Boffi, F. Brezzi, M. Fortin, Mixed Finite Element Methods and Applications, Vol. 44, Springer, 2013.

[27] R. J. LeVeque, Numerical methods for conservation laws, Vol. 214, Springer, 1992.

[28] D. Lilly, The representation of small-scale turbulence in numerical simulation experiments, in: Proceeding of the IBM Scientific Computing Symposium on Environmental Science, 1967, pp. 195–210.

[29] S. Micheletti, S. Perotto, Space-time adaptation for purely diffusive problems in an anisotropic framework, Int. J. Numer. Anal. Model. 7 (1) (2010) 125–155.

[30] J. M. Cascón, L. Ferragut, M. I. Asensio, Space-time adaptive algorithm for the mixed parabolic problem, Numer. Math. 103 (2006) 367–392.

[31] D. Meidner, B. Vexler, Adaptive space-time finite element methods for parabolic optimization problems, SIAM J. Control Optim. 46 (1) (2007) 116–142.

[32] M. Schmich, B. Vexler, Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations, SIAM J. Sci. Comput. 30 (1) (2008) 369–393.

[33] M. Morandi Cecchi, F. Marcuzzi, Adaptivity in space and time for shallow water equations, Int. J. Numer. Meth. Fluids 31 (1) (1999) 285–297.

[34] R. Verfürth, A Posteriori Error Estimation Techniques for Finite Element Methods, Oxford University Press, 2013.

[35] M. Ainsworth, J. T. Oden, A posteriori error estimation in finite element analysis, Pure and Applied Mathematics (New York), Wiley-Interscience [John Wiley & Sons], New York, 2000.

[36] W. Bangerth, R. Rannacher, Adaptive Finite Element Methods for Differential Equations, Springer Science & Business Media, 2003.

[37] O. C. Zienkiewicz, J. Z. Zhu, A simple error estimator and adaptive procedure for practical engineerng analysis, Int. J. Numer. Methods Eng. 24 (2) (1987) 337–357.

[38] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique, Int. J. Numer. Methods Eng. 33 (7) (1992) 1331–1364.

[39] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. part 2: Error estimates and adaptivity, Int. J. Numer. Methods Eng. 33 (7) (1992) 1365–1382.

[40] R. Rodríguez, Some remarks on Zienkiewicz-Zhu estimator, Numer. Methods Partial Differ. Equ. 10 (5) (1994) 625–635.

[41] M. Křížek, P. Neittaanmäki, Superconvergence phenomenon in the finite element method arising from averaging gradients, Numer. Math. 45 (1984) 105–116.

[42] G. Maisano, S. Micheletti, S. Perotto, C. L. Bottasso, On some new recovery-based a posteriori error estimators, Comput. Methods Appl. Mech. Eng. 195 (37-40) (2006) 4794–4815.

[43] X. D. Li, N. E. Wiberg, A posteriori error estimate by element patch post-processing, adaptive analysis in energy and L_2 norms, Comput. Struct. 53 (4) (1994) 907–919.

[44] S. Micheletti, S. Perotto, Anisotropic adaptation via a Zienkiewicz–Zhu error estimator for 2D elliptic problems, in: Numerical Mathematics and Advanced Applications, Springer, 2010, pp. 645–653.

[45] W. G. Habashi, M. Fortin, J. Dompierre, M. G. Vallet, Y. Bourgault, Anisotropic Mesh Adaptation: A Step Towards a Mesh-Independent and User-Independent CFD, Springer Netherlands, 1998, p. 99–117.

[46] Y. Belhamadia, A. Fortin, Y. Bourgault, On the performance of anisotropic mesh adaptation for scroll wave turbulence dynamics in reaction–diffusion systems, J. Comput. Appl. Math. 271 (2014) 233–246.

[47] S. Micheletti, S. Perotto, L. Soli, Topology optimization driven by anisotropic mesh adaptation: Towards a free-form design, Comput. Struct. 214 (2019) 60–72.

[48] M. Giacomini, S. Perotto, Anisotropic mesh adaptation for region-based segmentation accounting for image spatial information, Comput. Math. Appl. 121 (2022) 1–17.

[49] D. Cortellessa, N. Ferro, S. Perotto, S. Micheletti, Enhancing level set-based topology optimization with anisotropic graded meshes, Appl. Math. Comput. 447 (2023) 127903.

[50] N. Ferro, S. Perotto, D. Bianchi, R. Ferrante, M. Mannisi, Design of cellular materials for multiscale topology optimization: application to patient-specific orthopedic devices, Struct. Multidiscip. Optim. 65 (3) (2022) 79.

[51] P. J. Frey, P. L. George, Mesh Generation: Application to Finite Elements, John Wiley & Sons, 2008.

[52] S. Micheletti, S. Perotto, Reliability and efficiency of an anisotropic Zienkiewicz-Zhu error estimator, Comput. Methods Appl. Mech. Engrg. 195 (9–12) (2006) 799–835.

[53] E. Erturk, T. C. Corke, C. Gökçöl, Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers, Int. J. Numer. Meth. Fluids 48 (7) (2005) 747–774.

[54] E. Erturk, Numerical solutions of 2-D steady incompressible flow over a backward-facing step, part i: High Reynolds number solutions, Comput. Fluids 37 (6) (2008) 633–655.

[55] F. Hecht, New development in freefem++, J. Numer. Math. 20 (3-4) (2012) 251–266.

[56] H. C. Kuhlmann, F. Romanò, The lid-driven cavity, in: Computational modelling of bifurcations and instabilities in fluid dynamics, Vol. 50 of Comput. Methods Appl. Sci., Springer, Cham, 2019, pp. 233–309.

[57] S. Micheletti, S. Perotto, Output functional control for nonlinear equations driven by anisotropic mesh adaption: The Navier–Stokes equations, SIAM J. Sci. Comput. 30 (6) (2008) 2817–2854.

[58] T. Von Kármán, Mechanical Similitude and Turbulence, no. 611, National Advisory Committee for Aeronautics, 1931.

[59] L. Prandtl, Zur turbulenten strömung in rohren und längs platten, in: Ergebnisse der aerodynamischen Versuchsanstalt zu Göttingen Lfg. 4, De Gruyter, 1932, pp. 18–29.

[60] T. Von Karman, Turbulence and skin friction, J. Aeronaut. Sci. 1 (1) (1934) 1–20.

[61] M. Benjamin, S. P. Domino, G. Iaccarino, Neural networks for large eddy simulations of wall-bounded turbulence: numerical experiments and challenges, Eur. Phys. J. E. 46 (7) (2023) 55.

[62] S. T. Bose, G. I. Park, Wall-modeled large-eddy simulation for complex turbulent flows, Annu. Rev. Fluid Mech. 50 (1) (2018) 535–561.

[63] J. Husson, M. Terracol, S. Deck, T. Le Garrec, A comprehensive framework for robust hybrid RANS/LES simulations of wall-bounded flows in LBM, J. Comput. Phys. 502 (2024) Paper No. 112814, 23.

[64] Y. Bazilevs, C. Michler, V. M. Calo, T. J. R. Hughes, Weak Dirichlet boundary conditions for wall-bounded turbulent flows, Comput. Methods Appl. Mech. Eng. 196 (49-52) (2007) 4853–4862.

[65] Y. Bazilevs, T. J. R. Hughes, Weak imposition of Dirichlet boundary conditions in fluid mechanics, Comput. Fluids 36 (1) (2007) 12–26.

[66] Y. Bazilevs, C. Michler, V. M. Calo, T. Hughes, Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes, Comput. Methods Appl. Mech. Eng. 199 (13-16) (2010) 780–790.

[67] 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. Eng. 285 (2015) 379–405.

[68] J. Nikuradse, Strömungsgesetze in rauhen rohren vdi-forschungsheft (1933).

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

59/2024 Carbonaro, D.; Ferro, N.; Mezzadri, F.; Gallo, D.; Audenino, A.; Perotto, S.; Morbiducci, U.; Chiastra, C.
Easy-to-use formulations based on the homogenization theory for vascular stent design and mechanical characterization

58/2024 Ciaramella, G.; Vanzan, T.
Variable reduction as a nonlinear preconditioning approach for optimization problems

Speroni, G.; Ferro, N.
A novel metric-based mesh adaptation algorithm for 3D periodic domains

Temellini, E.; Ferro, N.; Stabile, G.; Delgado Avila, E.; Chacon Rebollo, T.; Perotto, S.
Space-time mesh adaptation for the VMS-Smagorinsky modeling of high Reynolds number flows

57/2024 Negrini, G.: Parolini, N.; Verani, M.
An Immersed Boundary Method for Polymeric Continuous Mixing

56/2024 Parolini, N.; Covello, V.; Della Rocca, A.; Verani, M.,
Design of a checkerboard counterflow heat exchanger for industrial applications

55/2024 Artoni, A.; Ciaramella, G.; Gander, M.J.; Mazzieri, I.
Schwarz Waveform Relaxation and the Unmapped Tent-Pitching Method in 3D

54/2024 Antonietti, P.F.; Botti, M.; Cancrini, A.; Mazzieri, I.
A polytopal discontinuous Galerkin method for the pseudo-stress formulation of the unsteady Stokes problem

53/2024 Caldana, M.; Hesthaven, J. S.
Neural ordinary differential equations for model order reduction of stiff systems

Negrini, G.; Parolini, N.; Verani, M.
Immersed Boundary Method for Polymeric Continuous Mixing