Adaptive control for iterative solvers in an FE framework with mesh adaptation, for CFD simulations of industrial processes

Gabriel Manzinali

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Mines Paris-PSL

Adaptive control for iterative solvers in an FE framework with mesh adaptation, for CFD simulations of industrial processes

Contrôle adaptatif pour les solveurs itératifs dans un cadre FE avec adaptation de maillage, pour les simulations CFD de processus industriels

Soutenue par

Gabriel Manzinali

Le 16/07/2021

Ecole doctorale n° 364

Sciences Fondamentales et Appliquées

Spécialité

Mathématiques numériques, Calcul intensif et Données

Composition du jury :

Alvaro COUTINHO Professeur, Federal University of Rio de Janeiro	*Président*
Suzanne Michelle SHONTZ Professeur, The University of Kansas	*Rapporteur*
Erwan LIBERGE Professeur, Université de La Rochelle	*Examinateur*
Elie, HACHEM Professeur, Mines Paris-PSL	*Directeur de thèse*
Yousef MESRI Chargé de Recherche, Mines Paris-PSL	*Examinateur*
Aurélien LARCHER Chargé de Recherche, Mines Paris-PSL	*Examinateur*

MINES PARIS | PSL

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

Résumé *L'objectif de ce travail est de proposer un critère d'arrêt pratique et général utilisant une approche a posteriori, qui s'appuie sur les estimations d'erreur disponibles à partir de la procédure d'adaptation du maillage. Ce critère d'arrêt doit être robuste et applicable aux différents types d'équations utilisées pour décrire la physique complexe impliquée dans un problème de transfert de chaleur conjugué. Le but final est de prouver qu'avec un tel critère d'arrêt il est possible de réduire drastiquement le temps CPU nécessaire à la résolution du système linéaire issu de la discrétisation Eléments Finis.*

1.1 General introduction

In the last decades Computational Fluid Dynamics (CFD) has become a widespread practice in several industrial fields, e.g., aerodynamics, aeroacoustic, turbomachinery. The pace of innovation seems to be speeding up and the process of innovation

itself is changing to follow this acceleration. A few decades ago any industrial product had a time to market of 2-3 years, while nowadays the design and production cycle would often take less than 18 months. This has a big impact on the technologies used to complete this process. Numerical simulations answered to the need of technologies that could give directions to follow in an early stage of the design process, saving the critical amount of time that is needed for prototyping and classic experimental tests. These technologies are not obsolete at all, but can be reserved to a more specific use, in an advanced stage of the project, and as a reference and calibration for the numerical methods. In the fluid dynamics field, a vast range of numerical application became available during the years. Available simulations techniques started from small stationary submodel simulations to more complex thermal coupling applications, possibly taking into account phase changes or fluid-structure interactions. The physics involved in all these kinds of problems are extremely challenging, and this led to a great academic interest in developing mathematical models and numerical methods to simulate these phenomena. The increasing computing power that is becoming available in the last decades is also increasing the attraction of this field of research. These technologies are becoming more and more accessible, and what was before at exclusive use of those companies that had access to a big computing resources, is nowadays accessible to everyone on his laptop. This leads to a constant need to improvement in the efficiency of the simulation process, with a specific focus on the accuracy required by the needs of the company.

One physical phenomenon that has great interest for industrial applications is the heat transfer that takes place when a solid is immersed in a fluid flow at a different temperature. Typical examples of the possible applications can be the cooling of specific parts of racing cars (e.g. wheels internals, engine) or aircraft, turbomachineries (e.g. turbine blades) [1], quenching of metal parts[2], heat treatments in industrial furnaces [3].

These types of phenomena typically involve very complex flow features with steep gradients. To allow a better representation of these features, one generally needs to have a finer discretisation of the computational domain, to have a better approximation of the solution. This achieved using a finer mesh, and leads to an increase in the computational demands. To deal with this inconvenience a new field of research arose, focusing on having a finer spatial discretization adapted to the region of the domain where it is required. The concept of mesh adaptation was introduced by Allgower in [4], and later developed in [5]. Mesh adaptation plays an important role in reducing the CPU time required by the simulation while ensuring high accuracy of the solution. The idea is to refine the mesh only in the regions where either the solution or its gradients have rapid variations, to have the best possible accuracy for a given number of mesh nodes. A vector or a scalar

function is used to monitor and control size, shape, and orientation of the elements of the new mesh. This function is usually designed to give an estimate of a suitable norm of the solution error, which is required to be equidistributed over the domain. Mesh adaptation can be described as the mathematical equivalent of a coordinate transformation. The idea of using an error estimator to drive the mesh adaptation procedure first appeared in [6, 7]. Where steep gradiends and directional feature are strongly characterizing the flow, however, isotropic mesh adaptation may lead to a big increase of the number of elements. In these situations, anisotropic meshes are a very interesting candidate to handle the directional aspect of a solution's variations. These types of meshes were first introduced in [8–10] in the framework of advancing front methods.

Given a certain level of accuracy obtained with the anisotropic mesh adaptation, the second index to evaluate an efficient simulation process is the CPU time needed. Simulating full scale models of industrial processing usually leads to huge computational requirements in terms of memory and time. This processes involve complex nonlinear physics, complex geometries and long characteristic periods. Even with a powerful workstation with 32 cores, it will take 5 days to simulate 5sec of a process that lasts 30min. This gives a glimpse of how it is important to have an efficient solver, to reduce as much as possible the time required for the solution. To impact this side of the problem, we should focus on the core of a numerical simulation code, and this is the iterative linear solver. For each numerical simulation, up to 60-80% of the CPU time is spent to solve iteratively the linear system that stems from the discretization of the governing equations. Therefore an optimal control of the iterative procedure and its convergence is critical to increase efficiency and performance of the simulation process.

When dealing with approximation errors introduced with numerical modeling, a typical approach is to consider the solution of the linear system at each level to be exact. However, for large problems, especially in three dimensions, the exact solution of the linear systems may be impossible to obtain efficiently, by using sparse direct solvers. In these cases one should adopt iterative linear solvers, which compute approximations to the exact finite element solution at each iteration of the linear solver. This leads to the need to estimate the error introduced, and control them to optimize the efficiency of the convergence procedure. The question of accuracy of iterative methods to solve a finite element system has been considered in [11, 12] where heuristic a priori stopping criteria for iterative methods are proposed. More recent contributions rely on a posterior approaches, with suitable estimates of the algebraic error contribution [13–17].

1.2 Objective of this work

The aim of this work is to propose a practical and general stopping criterion using an a posteriori approach, that relies on the error estimates available from the mesh adaptation procedure. This stopping criterion has to be robust and applicable to the different types of equations used to describe the complex physics involved in a conjugate heat transfer problem. The final goal is to prove that with such stopping criterion is possible to drastically reduce the CPU time required for the solution of the linear system that stems from the Finite Element discretization.

1.3 Working environment

This thesis work was developed at the Center for Material Forming, "CEntre de Mise En Forme des matériaux" (CEMEF, www.cemef.mines- paristech.fr) in the Sophia Antipolis technology park. The numerical implementation of the methods presented in this thesis work was made in the CIMLIB-CFD finite element library developed by the CFL team [18, 19]. This finite element library is developed as a fully parallel code in C++, an object oriented programming language, with the continuous contribution of the Ph.D. students, research assistants and professors of the team.

1.4 Layout of the thesis

The thesis is divided into five chapters. After a general introduction to the topic developed in this thesis, in Chapter 2. I introduce the general framework used to model the conjugate heat transfer in fluid flows. Chapter 3 is devoted to the presentation of the mesh adaptation procedure, with error estimators. In Chapter 4 we investigate stopping criteria for iterative solvers, and propose the adaptive stopping criterion for CDR and Navier-Stokes equations.

1.5 Author's contribution during the PhD

The contribution of the author in terms of publications, oral communications and prizes is presented below.

1.5.1 Journal articles

1. Manzinali G., Mesri Y., Hachem H. *Adaptive stopping criterion for iterative linear solvers in an AFEM framework, with applications to convection-*

dominated problems. Computer Methods in Applied Mechanics and Engineering, vol. 340, pp. 864-880, 2018.

1.5.2 Communications

1. Manzinali G., Mesri Y., Hachem H. *Anisotropic mesh adaptation and effects on the conditioning of unstructured finite element solvers*. 26th International Meshing Roundtable, IMR26, September 2017, Barcelona, Spain.

2. Manzinali G., Hachem H. *Adaptive Stopping Criterion for Iterative Linear Solvers in an Anisotropic Stabilized AFEM Framework*. 13th World Congress on Computational Mechanics, July 2018, New York, USA.

3. Manzinali G., Hachem H. *Adaptive Stopping Criterion for Iterative Linear Solvers in an Anisotropic AFEM Framework*. Fluids and Complexity, December 2018, Nice, France.

4. Manzinali G., Hachem H. *Adaptive stopping criterion for iterative solvers and efficient CPU time reduction, application to Navier-Stokes and thermal probems*. 20th International Conference on Fluid Flow Problems, April 2019, Chicago, USA.

1.5.3 Awards

1. Travel award from the IACM/USACM to attend at the International Conference on Fluid Flow Problems, April 2019, Chicago, USA.

1.6 Bibliography

[1] M. Hadziabdic, K. Hanjalic, Vortical structures and heat transfer in a round impinging jet, Journal of Fluid Mechanics 596 (2008) 221–260. 2

[2] C. Bahbah, Advanced numerical methods for the simulation of the industrial quenching process, Theses, Université PSL (Jan. 2020). 2

[3] E. Hachem, B. Rivaux, T. Kloczko, H. Digonnet, T. Coupez, Stabilized finite element method for incompressible flows with high reynolds number, Journal of Computational Physics 229 (23) (2010) 8643–8665. 2

[4] E. Allgower, S. McCormick, Newton's method with mesh refinements for numerical solution of nonlinear two-point boundary value problems, Numerische Mathematik 29 (3) (1978) 237–260. 2

[5] I. Babuška, O. Zienkiewicz, E. De Oliveira, Accuracy estimates and adaptive refinements in finite element computations, John Wiley & Sons, 1986. 2

[6] M. J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, Journal of Computational Physics 53 (3) (1984) 484–512. 3

[7] D. C. Arney, J. E. Flaherty, An adaptive mesh-moving and local refinement method for time-dependent partial differential equations, ACM Transactions on Mathematical Software (TOMS) 16 (1) (1990) 48–71. 3

[8] O. C. Zienkiewicz, J. Wu, Automatic directional refinement in adaptive analysis of compressible flows, International Journal for Numerical Methods in Engineering 37 (13) (1994) 2189–2210. 3

[9] R. Löhner, Adaptive remeshing for transient problems, Computer Methods in Applied Mechanics and Engineering 75 (1-3) (1989) 195–214.

[10] J. Peraire, M. Vahdati, K. Morgan, O. C. Zienkiewicz, Adaptive remeshing for compressible flow computations, Journal of Computational Physics 72 (2) (1987) 449–466. 3

[11] M. Arioli, D. Loghin, A. J. Wathen, Stopping criteria for iterations in finite element methods, Numerische Mathematik 99 (3) (2005) 381–410. 3

[12] P. Deuflhard, Cascadic conjugate gradient methods for elliptic partial differential equations: algorithm and numerical results, Contemporary Mathematics 180 (1994) 29–29. 3

[13] M. Picasso, A stopping criterion for the conjugate gradient algorithm in the framework of anisotropic adaptive finite elements, Communications in Numerical Methods in Engineering 25 (4) (2009) 339–355. 3

[14] A. Ern, M. Vohralík, Adaptive inexact newton methods with a posteriori stopping criteria for nonlinear diffusion pdes, SIAM Journal on Scientific Computing 35 (4) (2013) A1761–A1791.

[15] P. Jiránek, Z. Strakoš, M. Vohralík, A posteriori error estimates including algebraic error and stopping criteria for iterative solvers, SIAM Journal on Scientific Computing 32 (3) (2010) 1567–1590.

[16] M. Vohralík, M. F. Wheeler, A posteriori error estimates, stopping criteria, and adaptivity for two-phase flows, Computational Geosciences 17 (5) (2013) 789–812.

[17] D. A. Di Pietro, M. Vohralík, S. Yousef, An a posteriori-based, fully adaptive algorithm with adaptive stopping criteria and mesh refinement for thermal multiphase compositional flows in porous media, Computers & Mathematics with Applications 68 (12) (2014) 2331–2347. 3

[18] T. Coupez, Grandes transformations et remaillage automatique, Ph.D. thesis, Paris, ENMP (1991). 4

[19] H. Digonnet, T. Coupez, Object-oriented programming for "fast and easy" development of parallel applications in forming processes simulation, in: Computational Fluid and Solid Mechanics 2003, Elsevier, 2003, pp. 1922–1924. 4

Chapter 2

Conjugate heat transfer in fluid flows

Contents

Résumé *Dans ce chapitre, nous présentons le cadre principal pour la résolution numérique des écoulements de fluides et des problèmes de transfert de chaleur conjugués. Une discrétisation par éléments finis du domaine de calcul est adoptée pour sa précision et son efficacité avec les géométries complexes typiques des simulations industrielles. Cependant, la formulation Galerkin standard est affectée par les oscillations numériques dans les problèmes dominés par la convection ou la diffusion, en particulier en présence de gradients prononcés. Pour surmonter cette limitation, nous avons choisi la méthode Streamline Upwind Petrov–Galerkin (SUPG) appliquée aux problèmes de convection–diffusion–réaction dominés par la convection. Cette approche est efficace pour éliminer les instabilités en ajoutant une diffusion numérique dans la direction convective. Pour résoudre le problème d'écoulement de fluide, nous utilisons une approche multi-échelle variationnelle pour résoudre les équations incompressibles de Navier-Stokes. Cette méthode de stabilisation s'est avérée efficace pour gérer les problèmes dominés par la convection et traiter la condition de stabilité inf-sup. Lorsqu'il s'agit d'écoulements turbulents, nous avons introduit les équations RANS avec des modèles de turbulence, qui fournissent la solution d'écoulement de fluide moyen sans résoudre les petites échelles de turbulence. Tous ces ingrédients seront utilisés dans la suite dans le but de résoudre des problèmes industriels complexes de transfert de chaleur. Cependant, pour résoudre toutes ces équations pour des applications industrielles, il faut s'appuyer sur des procédures itératives. Le but de ce travail est de fournir un contrôle adaptatif pour les solveurs itératifs, afin de rendre la procédure de résolution plus rapide et plus efficace.*

2.1 Introduction

The study of conjugate heat transfer in fluid flows has been one of the most powerful tool in numerical applications to investigate natural and industrial phenomena in different areas ranging from meteorology to environment monitoring, from food processing to medical applications, from automotive and aerospace engines to design of heating and cooling process in material forming. Extensive work has been done, on both numerical and experimental sides, to develop more comprehensive numerical models for real life applications. This, backed by the latest improvements in computational performance and accessibility to the resources, is paving the path for a transition from an extensive use of experimental investigation on simplified problems to more fast and economical numerical analyses. Therefore comes the need to meet the expectation providing improved numerical modeling to achieve better accuracy and increased performances.

This chapter is centered on the modeling and solution methods for the heat transfer and fluid flows problems. The Navier–Stokes equations for fluid flows are solved coupled with the heat equation. Further treatment may be needed for high

Reynolds number flows, with the application of the so called Reynolds-Averaged Navier–Stokes equation, to provide an averaged version of the flow solution. In this case supplementary variables are introduced to model the turbulence phenomena, thus needing additional closure equations to be solved. The CIMLIB-CFD library, developed by the CFL group of Mines ParisTech, is used to solve the numerical problem. The methods implemented in the library have been developed and validated in previous works, and are built on a stabilized, anisotropic finite element discretization framework. For further detail we refer to [1–4].

In the following we present the general set of equations and stabilization methods that will be used in this work to solve heat transfer in industrial fluid flows problems.

2.2 Convection Diffusion Reaction equation

The Convection Diffusion Reaction equation (CDR) has always been of central importance for the numerical modeling of industrial applications. The continuous development of new numerical method for the solution of CDR equations, allowed to tackle more and more challenging configurations. Historically the first method applied to these type of equation was the Galerkin Finite Element method. Using this method however, global spurious oscillations arise in the areas of strong gradients in the solution, and in convection-dominated regimes. This oscillations affect first the accuracy of the solution and also the stability of the simulation. A variety of stabilization approaches have been proposed in the literature, with the most popular being the Streamline Upwind Petrov–Galerkin method proposed by Brooks and Hughes [5, 6] and revised by Scovazzi in [7]. This method substantially removes the oscillation in the standard Galerkin formulation maintaining the consistency of the problem. In this work we rely on this method for the stabilization of the CDR problems, combined with an anisotropic mesh adaptation procedure to capture sharp gradients in the solution.

In this section we start with a description of the equations used for the modeling of the problem, then we introduce the standard Galerkin formulation, followed by the SUPG technique needed for stabilization.

2.2.1 Governing equation

We consider the following problem that models the transport of a quantity u through convection diffusion and reaction, posed on $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, a bounded polyhedral

domain with boundary Γ:

$$\begin{cases} \partial_t u + \boldsymbol{v} \cdot \boldsymbol{\nabla} u - \kappa \boldsymbol{\Delta} u + \sigma u = f & \text{in } \Omega \times (0,T) \ , \\ u(\,.\,,0) = u_0 & \text{in } \Omega \ , \\ u = g & \text{on } \Gamma \ , \end{cases} \tag{2.1}$$

with $\boldsymbol{v} \in (\mathrm{W}^{1,\infty}(\Omega))^d$ a divergence-free advective field, κ a diffusion coefficient, $\sigma \in \mathrm{L}^2(\Omega)$ a reaction coefficient, and $f \in \mathrm{L}^2(\Omega)$ a given source term; additionally are provided u_0 the initial data, and g a given function prescribed at the boundary. This type of problem can exhibit four types of boundary layers, depending on the relative velocity direction [8]:

- Regular boundary layers: appear at the outflow boundary where the velocity field $\boldsymbol{v}$ is not parallel to Γ, with $\boldsymbol{v} \cdot \boldsymbol{n} < 0$.
- Parabolic boundary layers: appear at the characteristic boundary where the velocity field $\boldsymbol{v}$ is parallel to Γ, they are thicker then the regular boundary layer.
- Corner boundary layers: appear at the corners of the domain Ω where two boundary layers intersect each other.
- Interior layers: appear due to discontinuities in the boundary conditions at the inflow boundary, with these discontinuities propagated across the domain following the direction of $\boldsymbol{v}$.

2.2.2 Standard Galerkin formulation

We define a weak problem involving the usual function spaces L^2 and H^1, with L^2 the Hilbert space defined by

$$\mathrm{L}^2(\Omega) = \left\{ v : \int_\Omega |v(\boldsymbol{x})|^2 \, \mathrm{d}\boldsymbol{x} < \infty \right\} \ , \tag{2.2}$$

and $\mathrm{H}^1 \in \mathrm{L}^2$ the Sobolev space of functions with square integrable first order derivatives

$$\mathrm{H}^1(\Omega) = \{ v \in \mathrm{L}^2(\Omega) : \| \boldsymbol{\nabla} v \| \ \in \ \mathrm{L}^2(\Omega) \} \ . \tag{2.3}$$

The weak formulation is obtained by multiplying Equation (2.1) by an appropriate test function $w \in \mathrm{H}^1_0(\Omega)$ and by integrating over the computational domain. The weak formulation of the convection–diffusion–reaction problem reads

$$(\partial_t u, w) + \mathcal{B}(u,w) = \mathcal{F}(w) \quad \forall w \in \mathrm{H}^1_0(\Omega) \ , \tag{2.4}$$

where $(u, w) \equiv \int_\Omega u(\boldsymbol{x})\, v(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$ denotes the L^2 inner product, and

$$\begin{cases} \mathcal{B}(u,w) &= (\boldsymbol{v}\cdot\boldsymbol{\nabla}u, w) - (\kappa\boldsymbol{\nabla}u, \boldsymbol{\nabla}w) + (\sigma u, w)\ , \\ \mathcal{F}(w) &= (f, w)\ , \end{cases} \tag{2.5}$$

accordingly, the solution will be sought in $\mathrm{H}^1(\Omega)$ such that $u = g$ on Γ.

For the discretization in space we consider a finite element admissible mesh $\mathcal{T}_h$ on the computational domain Ω. In this framework the function space $\mathrm{H}^1_0(\Omega)$ is approximated by the finite dimensional space $\mathcal{V}_h \subset \mathrm{H}^1_0(\Omega)$. In the following $\mathcal{V}_h$ is chosen as the space of piecewise linear functions on the elements $K \in \mathcal{T}_h$. The Galerkin finite element formulation (2.6) reads

$$(\partial_t u_h, w_h) + \mathcal{B}(u_h, w_h) = \mathcal{F}(w_h) \quad \forall w_h \in \mathcal{V}_h\ , \tag{2.6}$$

When convection dominates diffusion, the solution of the problem becomes challenging. In these conditions when the Peclet number local to an element

$$Pe|_K = \frac{\|\boldsymbol{v}\|_K h_K}{2a}$$

with the element size $h_K = \mathrm{diam}(K)$, becomes greater than one, the solution behavior varies across the domain: in some regions the solution is smooth, in others the solution exhibits high gradients, developing boundary layers. Since the standard Galerkin finite element discretization is a centered approximation, it is well-known that node-to-node oscillations would possibly appear, spoiling the global accuracy of the solution. Upwind techniques, such as stabilized finite elements, can be used to solve this issue.

2.2.3 Streamline Upwind Petrov–Galerkin stabilization

The SUPG stabilization applied to (2.6) introduces an additional contribution to the standard Galerkin test functions w_h in the upwind direction and for all terms in the equation. Using this method more weight is added to the nodes in the upstream direction reducing the weight of the nodes in the downstream direction. The modified test function is defined by

$$\tilde{w}_h = w_h + \tau_K \boldsymbol{v}\cdot\boldsymbol{\nabla}w_h\ . \tag{2.7}$$

with τ_K a piecewise-defined coefficient to be defined. The weak formulation (2.6) taking into account SUPG terms reads

$$(\partial_t u_h, w_h) + \mathcal{B}_\tau(u_h, w_h) = \mathcal{F}_\tau(w_h) \quad \forall w_h \in \mathcal{V}_h\ , \tag{2.8}$$

where

$$\begin{cases} \mathcal{B}_\tau(u_h, w_h) &= \mathcal{B}(u_h, w_h) + \sum\limits_{K \in \mathcal{T}_h} \tau_K(\boldsymbol{v} \cdot \boldsymbol{\nabla} u_h - \kappa \boldsymbol{\Delta} u_h + \sigma u_h, \boldsymbol{v} \cdot \boldsymbol{\nabla} w_h)_K \\ \mathcal{F}_\tau(w_h) &= \mathcal{F}(w_h) + \sum\limits_{K \in \mathcal{T}_h} \tau_K(f, \boldsymbol{v} \cdot \boldsymbol{\nabla} w_h)_K \ , \end{cases} \tag{2.9}$$

with $(u, v)_K = \int_K u(\boldsymbol{x})\, v(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$ the L^2 inner-product on each element $K \in \mathcal{T}_h$. This formulation is equivalent to introducing a local diffusion in the streamline direction of convection. The additional stabilizing terms vanish for the exact solution of the problem, to the limit of the continuous problem.

The SUPG method used in this work is globally stable, and is a good compromise between stability and accuracy of the solution. If the solution exhibits steep layers however, it is impossible to achieve optimal orders of convergence i.e. second order for the L^2 norm and first order for the H^1 norm. In our framework the anisotropic mesh adaptation technique is used to help recover this loss in convergence optimality. For the details on the choice of the stabilization parameter τ_K we refer to [2] and references therein.

2.2.4 Time discretization scheme

The problem defined by Equation (2.8) yields the system of first order differential equations:

$$\mathrm{C}\dot{\mathbf{u}} + \mathrm{K}_\tau \mathbf{u} = \mathbf{f}_\tau \ , \tag{2.10}$$

where $\mathbf{u}$ is the vector of nodal values of the solution, C the mass matrix, K_τ the matrix of discrete operator $\mathcal{B}_\tau$, and $\mathbf{f}_\tau$ the source term. The system of Ordinary Differential Equations (2.10) has to be integrated in time. Using the θ time discretization schemes, the derivative of the solution with respect to time can be approximated at time t^n by:

$$\mathrm{C}\frac{\mathbf{u}^n - \mathbf{u}^{n-1}}{\Delta t} + \mathrm{K}_\tau \left(\theta \mathbf{u}^n + (1-\theta)\mathbf{u}^{n-1}\right) = \mathbf{f}_\tau \ , \tag{2.11}$$

where Δt is the time-step, $n = 1, \ldots, N$, and $0 \leqslant \theta \leqslant 1$. We recall that this family includes the backward Euler scheme ($\theta = 1$), the Crank–Nicolson scheme ($\theta = 0.5$) and the forward Euler scheme ($\theta = 0$), and that the source term may also be evaluated in the same manner.

2.3 The incompressible Navier–Stokes equations

The incompressible homogeneous Navier–Stokes equations in isothermal equilibrium are a system of Partial Differential Equations describing the evolution of fluid flows

through one vector variable, the velocity $\boldsymbol{v}$, and one scalar variable, the pressure p. Named after Claude-Louis Navier and George Gabriel Stokes, they remain one of the most important open problems in mathematics. Despite the lack of proof of the existence and smoothness of solutions, these equations are of great importance for a wide range of scientific and engineering applications. They can be used to model problems ranging from aerospace and automotive aerodynamics to hemodynamics in biomedical applications, from meteorology to complex fluids such as foam, suspensions and liquid crystals. Due to this large field of applications a big effort has been put to devise accurate and robust numerical methods to solve these equations. However the heterogeneity of the behavior at different scales still poses a great challenge to achieve accurate solutions.

In this section we present the mathematical formulation of the equations governing incompressible flows. The incompressibility hypothesis is introduced when the effects of pressure on the fluid density are negligible. We consider a stabilized finite element framework to deal with the instabilities that corrupts the standard Galerkin scheme. The causes of these instabilities are found in the non-linear term in convective dominated regimes, and at the discrete level in space incompatibility between the velocity and the pressure approximation spaces. The inf-sup condition [9, 10], known as the Babuška–Brezzi condition, is not satisfied by a standard Galerkin formulation with equal-order approximation spaces. In the literature several solution were proposed to deal with this instabilities. The Mini elements method [11] is a popular mixed formulation using the same order of approximation for pressure and velocity spaces, but enriching the latter with bubble functions that vanish at the element boundaries. This formulation, however, preforms poorly in convection-dominated problems. To overcome this limitation, Franca in [12] proposed a residual-free bubble method to provide the stabilization terms. Residual-based stabilization methods were introduced in several works by Brezzi et al. [13] and Codina [14–16]. In this work we consider the multiscale approach, first introduced by Hughes in [17–19], with the formulation implemented and validated in the CIMLIB-CFD library by Hachem [3].

2.3.1 Governing equations

Let $\Omega \subset \mathbb{R}^d$ be the bounded fluid domain of dimension $d \in \{2, 3\}$, $\partial\Omega$ its boundary, and $(0, T)$ be the time interval, the transient incompressible Navier–Stokes equations in strong form read

$$\begin{cases} \rho\left[\partial_t \boldsymbol{v} + (\boldsymbol{v} \cdot \boldsymbol{\nabla})\boldsymbol{v}\right] - \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} = \boldsymbol{f} & \text{in } \Omega \times (0,T)\ , \\ \boldsymbol{\nabla}\cdot\boldsymbol{v} = 0 & \text{in } \Omega \times (0,T)\ , \end{cases} \tag{2.12}$$

with $\boldsymbol{v}$ the velocity, p the pressure, ρ the density assumed to be a positive constant real number, and $\boldsymbol{f}$ a given source term of finite energy representing external forces

acting on the fluid. The Cauchy stress tensor for a Newtonian fluid is given by

$$\boldsymbol{\sigma} = 2\mu\boldsymbol{\varepsilon}(\boldsymbol{v}) - p\mathbb{I} \ , \tag{2.13}$$

where μ is a positive real number called dynamic viscosity, and $\mathbb{I}$ the d-dimensional identity tensor. The strain rate tensor $\boldsymbol{\varepsilon}(\boldsymbol{v})$ defines the symmetric part of the velocity gradient:

$$\boldsymbol{\varepsilon}(\boldsymbol{v}) = \frac{1}{2}\left[\boldsymbol{\nabla v} + (\boldsymbol{\nabla v})^T\right] \tag{2.14}$$

In order to close the problem, Equations (2.12) are subject to appropriate initial and boundary conditions:

$$\begin{cases} \boldsymbol{v}(\boldsymbol{x}, 0) = \boldsymbol{v}_0(\boldsymbol{x}) & \text{in } \Omega \ , \\ \boldsymbol{v}(\cdot, t) = \boldsymbol{v}_D & \text{in } \Gamma_D \times (0, T) \ , \\ \boldsymbol{\nabla v} \cdot \boldsymbol{n} = \phi_N & \text{in } \Gamma_N \times (0, T) \ , \end{cases} \tag{2.15}$$

where $\partial\Omega = \Gamma = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \varnothing$. When prescribing Dirichlet boundary conditions, pressure is defined as a gauge pressure, so we can impose a condition $\int_\Omega p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 0$ as reference.

2.3.2 Galerkin finite elements formulation

To derive the weak formulation of Problem (2.12) we use the following subspaces of the usual Lebesgue function space of square integrable functions, and Hilbert space defined on Ω:

$$\begin{aligned} \mathcal{V} &= \{\boldsymbol{v} \in \left(\mathrm{H}^1(\Omega)\right)^d : \boldsymbol{v} = \boldsymbol{v}_D \text{ on } \Gamma_D\} \ , \\ \mathcal{V}^0 &= \{\boldsymbol{v} \in \left(\mathrm{H}^1(\Omega)\right)^d : \boldsymbol{v} = 0 \text{ on } \Gamma_D\} \ , \\ \mathcal{Q} &= \{q \in \mathrm{L}^2(\Omega)\} \ . \end{aligned} \tag{2.16}$$

We multiply the Equations in (2.12) by test functions $(\boldsymbol{w}, q) \in (\mathcal{V}^0, \mathcal{Q})$. Integrating by parts the viscous and pressure terms, an additional integral over the boundary Γ_N emerges. The weak form of Equations (2.12) reads:

$$\left|\begin{array}{ll} \text{Find } (\boldsymbol{v}, p) \in \mathcal{V} \times \mathcal{Q} \text{ such that:} & \\ (\rho\left[\partial_t \boldsymbol{v} + (\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{v}\right], \boldsymbol{w}) + & \\ \quad (2\mu\boldsymbol{\varepsilon}(\boldsymbol{v}), \boldsymbol{\varepsilon}(\boldsymbol{w})) - (p, \boldsymbol{\nabla}\cdot\boldsymbol{w}) = (\boldsymbol{f}, \boldsymbol{w}) + (\phi_N, \boldsymbol{w})_{\Gamma_N} & ,\forall\ \boldsymbol{w} \in \mathcal{V}^0 \\ (\boldsymbol{\nabla}\cdot\boldsymbol{v}, q) = 0 & ,\forall\ q \in \mathcal{Q} \end{array}\right. \tag{2.17}$$

with $(u, v)_{\Gamma_N} = \int_{\Gamma_N} u(\mathrm{s})v(s)\,\mathrm{d}s$, $\mathrm{d}s$ denoting the $d-1$ dimensional Lebesgue measure.

For the spatial discretization we consider the finite element admissible partition $\mathcal{T}_h$ of the domain Ω into a family simplex elements $\{K\}$. The previously defined function spaces $\mathcal{V}$ and $\mathcal{V}^0$ are approximated by the following discrete spaces with piecewise linear functions

$$\begin{aligned} \mathcal{V}_h &= \{\boldsymbol{v}_h \in \left(\mathrm{C}^0(\Omega)\right)^d : \boldsymbol{v}_h|_K \in \mathbb{P}_1(K)^d, \forall K \in \mathcal{T}_h\} \ , \\ \mathcal{V}_h^0 &= \{\boldsymbol{w}_h \in \mathcal{V}_h : \boldsymbol{w}_h|_\Gamma = 0\} \ , \\ \mathcal{Q}_h &= \{q_h \in \mathrm{C}^0(\Omega) : q_h|_K \in \mathbb{P}_1(K), \forall K \in \mathcal{T}_h\} \ . \end{aligned} \tag{2.18}$$

Therefore the Galerkin formulation of Problem (2.17) reads

$$\begin{aligned} &\text{Find } (\boldsymbol{v}_h, p_h) \in \mathcal{V}_h \times \mathcal{Q}_h \text{ such that:} \\ &(\rho\left[\partial_t \boldsymbol{v}_h + (\boldsymbol{v}_h \cdot \boldsymbol{\nabla})\boldsymbol{v}_h\right], \boldsymbol{w}_h) + \\ &\quad (2\mu\boldsymbol{\varepsilon}(\boldsymbol{v}_h), \boldsymbol{\varepsilon}(\boldsymbol{w}_h)) - (p_h, \boldsymbol{\nabla}\cdot \boldsymbol{w}_h) = (\boldsymbol{f}, \boldsymbol{w}_h) + (\phi_N, \boldsymbol{w}_h)_{\Gamma_N} \quad , \forall\ \boldsymbol{w}_h \in \mathcal{V}_h^0 \\ &(\boldsymbol{\nabla}\cdot \boldsymbol{v}_h, q_h) = 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad , \forall\ q_h \in \mathcal{Q}_h \end{aligned} \tag{2.19}$$

As we mentioned above the standard Galerkin method with P1/P1 elements (i.e. the same piecewise linear space for $\mathcal{V}_h$ and $\mathcal{Q}_h$) is not stable, because the inf-sup condition is not satisfied. Moreover, convection-dominated problems (i.e. problems where the convection term $(\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{v}$ is much larger than the diffusion term $\boldsymbol{\nabla}\cdot\left(2\mu\ \boldsymbol{\varepsilon}(\boldsymbol{v})\right)$) also lead to a loss of coercivity in formulation (2.17). In this work, we use a Variational MultiScale method which circumvents both problems through a Petrov–Galerkin approach. The basic idea is to consider that the unknowns can be split into two components, a coarse one and a fine one, corresponding to different scales or levels of resolution. First, we solve the fine scales in an approximate manner and then we replace their effect into the large-scale equation.

2.3.3 Variational Multiscale approach (VMS)

The VMS approach provides a natural stabilization by an orthogonal decomposition of the solution function spaces. We apply this decomposition to the velocity and pressure spaces

$$\begin{aligned} \tilde{\mathcal{V}} &= \mathcal{V}_h \oplus \mathcal{V}' \\ \tilde{\mathcal{Q}} &= \mathcal{Q}_h \oplus \mathcal{Q}' \ . \end{aligned} \tag{2.20}$$

Velocity and pressure can be split now into resolvable coarse-scale and unresolved fine-scale components

$$\begin{aligned} \tilde{\boldsymbol{v}} &= \boldsymbol{v}_h + \boldsymbol{v}' \\ \tilde{p} &= p_h + p' \ , \end{aligned} \tag{2.21}$$

with the fine scales that will provide stabilization where the solution will present steep gradients. The same decomposition is used for the test functions

$$\begin{aligned} \tilde{\boldsymbol{w}} &= \boldsymbol{w}_h + \boldsymbol{w}' \\ \tilde{q} &= q_h + q' \ , \end{aligned} \tag{2.22}$$

Using this decomposition the Problem (2.17) becomes

$$\left| \begin{array}{l} \text{Find } (\tilde{\boldsymbol{v}}, \tilde{p}) \in \tilde{\mathcal{V}} \times \tilde{\mathcal{Q}} \text{ such that:} \\ (\rho \left[\partial_t(\boldsymbol{v}_h + \boldsymbol{v}') + ((\boldsymbol{v}_h + \boldsymbol{v}') \cdot \boldsymbol{\nabla})(\boldsymbol{v}_h + \boldsymbol{v}')\right], (\boldsymbol{w}_h + \boldsymbol{w}')) \\ \qquad + (2\mu\boldsymbol{\varepsilon}(\boldsymbol{v}_h + \boldsymbol{v}'), \boldsymbol{\varepsilon}(\boldsymbol{w}_h + \boldsymbol{w}')) \\ \qquad - ((p_h + p'), \boldsymbol{\nabla}\cdot(\boldsymbol{w}_h + \boldsymbol{w}')) = (\boldsymbol{f}, (\boldsymbol{w}_h + \boldsymbol{w}')) + (\phi_N, (\boldsymbol{w}_h + \boldsymbol{w}'))_{\Gamma_N} \ , \\ (\boldsymbol{\nabla}\cdot(\boldsymbol{v}_h + \boldsymbol{v}'), (q_h + q')) = 0 \ , \\ \text{for any pair } (\boldsymbol{w}_h, q_h) \in \tilde{\mathcal{V}} \times \tilde{\mathcal{Q}}. \end{array} \right. \tag{2.23}$$

The function spaces of large and small scales have to be linearly independent for consistency, and the formulation of Problem (2.23) is linear with respect to the test function. Therefore we can dissociate the large and the small scales into two sub-problems:

Coarse-scale problem

$$\left| \begin{array}{l} (\rho \left[\partial_t(\boldsymbol{v}_h + \boldsymbol{v}') + ((\boldsymbol{v}_h + \boldsymbol{v}') \cdot \boldsymbol{\nabla})(\boldsymbol{v}_h + \boldsymbol{v}')\right], \boldsymbol{w}_h) \\ \qquad + (2\mu\boldsymbol{\varepsilon}(\boldsymbol{v}_h + \boldsymbol{v}'), \boldsymbol{\varepsilon}(\boldsymbol{w}_h)) \\ \qquad - ((p_h + p'), \boldsymbol{\nabla}\cdot\boldsymbol{w}_h) = (\boldsymbol{f}, \boldsymbol{w}_h) + (\phi_N, \boldsymbol{w}_h)_{\Gamma_N} \ , \\ (\boldsymbol{\nabla}\cdot(\boldsymbol{v}_h + \boldsymbol{v}'), q_h) = 0 \ . \end{array} \right. \tag{2.24}$$

Fine-scale problem

$$\left| \begin{array}{l} (\rho \left[\partial_t(\boldsymbol{v}_h + \boldsymbol{v}') + ((\boldsymbol{v}_h + \boldsymbol{v}') \cdot \boldsymbol{\nabla})(\boldsymbol{v}_h + \boldsymbol{v}')\right], \boldsymbol{w}') \\ \qquad + (2\mu\boldsymbol{\varepsilon}(\boldsymbol{v}_h + \boldsymbol{v}'), \boldsymbol{\varepsilon}(\boldsymbol{w}')) + \\ \qquad - ((p_h + p'), \boldsymbol{\nabla}\cdot\boldsymbol{w}') = (\boldsymbol{f}, \boldsymbol{w}') + (\phi_N, \boldsymbol{w}')_{\Gamma_N} \ , \\ (\boldsymbol{\nabla}\cdot(\boldsymbol{v}_h + \boldsymbol{v}'), q') = 0 \ . \end{array} \right. \tag{2.25}$$

A static condensation can be applied, where an approximate solution of the fine scales is plugged back into the coarse-scale problem. The effect of the fine-scales

approximation on the coarse-scale problem acts as a local stabilization, without being explicitly computed. To do so, we rearrange Equation (2.25) in the form

$$\left|\begin{array}{l}(\rho\partial_t \boldsymbol{v}', \boldsymbol{w}') + (\rho((\boldsymbol{v}_h + \boldsymbol{v}')\cdot\boldsymbol{\nabla})\boldsymbol{v}', \boldsymbol{w}')) \\ \qquad + (2\mu\boldsymbol{\varepsilon}(\boldsymbol{v}'), \boldsymbol{\varepsilon}(\boldsymbol{w}')) + (\boldsymbol{\nabla}p', \boldsymbol{w}') = (\mathcal{R}_M, \boldsymbol{w}') \\ (\boldsymbol{\nabla}\cdot\boldsymbol{v}', q') = (\mathcal{R}_C, q')\ ,\end{array}\right. \tag{2.26}$$

where $(\mathcal{R}_M, \boldsymbol{w}')$ and $(\mathcal{R}_C, q')$ are respectively residuals of the momentum and continuity large-scales projected onto the fine-scale spaces

$$\mathcal{R}_M = \boldsymbol{f} - \rho\partial_t \boldsymbol{v}_h - \rho(\boldsymbol{v}_h\cdot\boldsymbol{\nabla})\boldsymbol{v}_h - \boldsymbol{\nabla}p_h \tag{2.27}$$

$$\mathcal{R}_C = -\boldsymbol{\nabla}\cdot\boldsymbol{v}_h\ . \tag{2.28}$$

The next step into approximating the fine-scale problem is the introduction of several assumption to simplify the equations in terms of time-dependency and non-linearity:

- To reduce the computational cost of tracking the subscales in time, we drop the time dependency term in Equation (2.25). Note than some time dependency is retained in the large-scale residual term $\mathcal{R}_M$, see [20].
- As the large-scale gradients are dominant compared to the small scales, the convective velocity of the non-linear term may be approximated using only large-scale part

$$((\boldsymbol{v}_h + \boldsymbol{v}')\cdot\boldsymbol{\nabla})(\boldsymbol{v}_h + \boldsymbol{v}') \approx (\boldsymbol{v}_h\cdot\boldsymbol{\nabla})(\boldsymbol{v}_h + \boldsymbol{v}')$$

- As the fine-scale space is assumed to be orthogonal to the finite element space, the crossed viscous terms vanish in 2.24 and 2.25, [21];

The fine-scale problem will then reduce to the following

$$\left|\begin{array}{l}(\rho(\boldsymbol{v}_h\cdot\boldsymbol{\nabla})\boldsymbol{v}', \boldsymbol{w}') + (2\mu\boldsymbol{\varepsilon}(\boldsymbol{v}') : \boldsymbol{\varepsilon}(\boldsymbol{w}')) + (\boldsymbol{\nabla}p', \boldsymbol{w}') = (\mathcal{R}_M, \boldsymbol{w}') \\ (\boldsymbol{\nabla}\cdot\boldsymbol{v}', q') = (\mathcal{R}_C, q')\ ,\end{array}\right. \tag{2.29}$$

The fine-scale equation, however, is not trivial to be solved, so one possible way could be to approximate these subscales locally using the residual-based terms and stabilization parameters τ_v and τ_p as follows

$$\begin{aligned}\boldsymbol{v}' &= \tau_v \mathcal{R}_M \\ p' &= \tau_p \mathcal{R}_C\end{aligned} \tag{2.30}$$

Several definitions can be found in the literature for the stabilization parameters; Codina in [14] proposed the following stabilizing coefficients defined piecewise constant on each $K \in \mathcal{T}_h$:

$$\begin{aligned} \tau_v|_K &= \left[\left(\frac{c_1 \mu}{\rho h_m^2} \right)^2 + \left(\frac{c_2 \rho \|\boldsymbol{v}_h\|_K}{h_K} \right)^2 \right]^{-\frac{1}{2}} \\ \tau_p|_K &= \left[\left(\frac{\mu}{\rho} \right)^2 + \left(\frac{c_2 \rho \|\boldsymbol{v}_h\|_K}{c_1 h_K} \right)^2 \right]^{\frac{1}{2}} , \end{aligned} \tag{2.31}$$

where h_K is the characteristic length of the element K, $\|\boldsymbol{v}_h\|_K$ a characteristic norm of $\boldsymbol{v}_h$ in K and c_1, c_2 two constants independent on h_K. We assign these constants to $c_1 = 4$ and $c_2 = 2$ for linear elements, see [16]. Another possible way is to include the time step size of the temporal discretization in the expression of τ_v. This improves the convergence behavior of the algorithm to deal with the non-linearity of the problem.

Taking into account the assumptions on the fine-scales and substituting the expressions of both the fine-scale pressure and the fine-scale velocity of Relations (2.30), the coarse-scale system becomes

$$\left| \begin{aligned} &(\rho \partial_t \boldsymbol{v}_h, \boldsymbol{w}_h) + (\rho(\boldsymbol{v}_h \cdot \boldsymbol{\nabla})\boldsymbol{v}_h, \boldsymbol{w}_h) \\ &\qquad + (2\mu \boldsymbol{\varepsilon}(\boldsymbol{v}_h), \boldsymbol{\varepsilon}(\boldsymbol{w}_h)) - \sum_{K \in \mathcal{T}_h} (\tau_v \mathcal{R}_M, \rho(\boldsymbol{v}_h \cdot \boldsymbol{\nabla})\boldsymbol{w}_h)_K \\ &\qquad - (p_h, \boldsymbol{\nabla} \cdot \boldsymbol{w}_h) + \sum_{K \in \mathcal{T}_h} (\tau_p \mathcal{R}_C, \boldsymbol{\nabla} \cdot \boldsymbol{w}_h)_K = (f, \boldsymbol{w}_h) \\ &(\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, q_h) - \sum_{K \in \mathcal{T}_h} (\tau_v \mathcal{R}_M, \boldsymbol{\nabla} q_h) = 0 \ . \end{aligned} \right. \tag{2.32}$$

Finally, for the sake of completeness, we plug in the residuals explicit formulas in the coarse-scale problem to obtain

$$\left| \begin{aligned} &(\rho(\partial_t \boldsymbol{v}_h + (\boldsymbol{v}_h \cdot \boldsymbol{\nabla})\boldsymbol{v}_h), \boldsymbol{w}_h) + (2\mu \boldsymbol{\varepsilon}(\boldsymbol{v}_h), \boldsymbol{\varepsilon}(\boldsymbol{w}_h)) - (p_h, \boldsymbol{\nabla} \cdot \boldsymbol{w}_h) - (f, \boldsymbol{w}_h) \\ &+ \sum_{K \in \mathcal{T}_h} \tau_v \left(\rho \left(\partial_t \boldsymbol{v}_h + (\boldsymbol{v}_h \cdot \boldsymbol{\nabla})\boldsymbol{v}_h \right) + \boldsymbol{\nabla} p_h - f, \rho(v_h \cdot \boldsymbol{\nabla})\boldsymbol{w}_h \right)_K \\ &+ \sum_{K \in \mathcal{T}_h} \tau_v \left(\rho \left(\partial_t \boldsymbol{v}_h + (\boldsymbol{v}_h \cdot \boldsymbol{\nabla})\boldsymbol{v}_h \right) + \boldsymbol{\nabla} p_h - f, \boldsymbol{\nabla} q_h \right)_K \\ &+ (\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, q_h) + \sum_{K \in \mathcal{T}_h} \tau_p \left(\boldsymbol{\nabla} \cdot \boldsymbol{v}_h, \boldsymbol{\nabla} \cdot \boldsymbol{w}_h \right)_K = 0 \ . \end{aligned} \right. \tag{2.33}$$

The new formulation of Problem (2.33) can now be decomposed into four main terms:

- the standard Galerkin formulation
- fine-scale velocity upwind stabilization term
- fine-scale pressure stabilization term
- fine-scale velocity grad-div stabilization term.

All the added terms provide additional stabilization in convection-dominated regimes and allow to avoid instabilities caused by incompatible approximation spaces. Equations (2.33) are discretized in time by a semi-implicit scheme. The convective term, the viscous term and the pressure term in the momentum equation, as well the divergence term in the continuity equation, are integrated implicitly through a backward Euler scheme. All other contributions (i.e. the source term and the stabilization terms) are integrated explicitly by a forward Euler scheme.

One important parameter to address is the element characteristic length h_K, that significantly influences the behavior of the stabilization parameter. A common choice would be to define it as the minimum element size. When dealing with strongly anisotropic meshes, however, the definition of h_K is still an open problem and plays a critical role in the design of the stabilizing coefficients [14]. In this work we adopt the definition proposed in [22] to compute h_K. It consists in computing h_K as the diameter of the element in the direction of the velocity field v_h

$$h_K = \frac{2\|\boldsymbol{v}_h\|_K}{\sum_{\varphi_i} |(\boldsymbol{v}_h \cdot \boldsymbol{\nabla}\varphi_i)(\boldsymbol{x}_K)|} \tag{2.34}$$

with $\mathbb{P}_1(K) = \text{span}\{\varphi_i\}$ and $\boldsymbol{x}_K$ the cell centroid.

2.4 Turbulence modeling for Reynolds-Averaged Navier–Stokes equations

Turbulent flow dynamics are nowadays one of the most difficult challenges in fluid dynamics. Turbulence phenomena have been observed since the 16th century by Leonardo da Vinci, but it is only in the late 19th century that Osborne Reynolds demonstrated the transition from laminar to turbulent flow. In a classic experiment he examined the behavior of water flow under different flow rates using a small jet of dyed water introduced into the center of flow in a larger pipe. From the observations he made in these experiments he introduced the nondimensional number, named after himself, that characterizes this behavior through the ratio of inertial forces to viscous forces. In later years several studies aimed at describing the turbulence phenomena, showing the energy cascade from bigger eddies to the

smaller ones. A pioneering work done by Andrey Kolmogorov in the 1940s theorized the wide span of scales involved, from a macroscale at which the kinetic energy is supplied, to a microscale at which energy is dissipated by viscosity. Kolmogorov hypothesized that when these scales are well separated, the intermediate range of length scales would be statistically isotropic, and that its characteristics in equilibrium would depend only on the rate at which kinetic energy is dissipated at the small scale. The wide scale range and the chaotic fluctuations in the flow features led to numerous studies on the most suitable numerical method to tackle these type of problems. A straightforward approach would be to solve directly the Navier–Stokes equations to the Kolmogorov dissipative scales. However, his approach, known as Direct Numerical Simulation, is highly computationally expensive and can be used only with small problems and/or moderate Reynolds numbers. Other approaches consist in modeling the dynamics in part or all of the turbulent subscales. The [Very] Large Eddy Simulation approach solves only the large scale structures while modeling the effects due to the small scales, being still considerably computationally expensive. A more industrially relevant approach the Reynolds-Averaged Navier–Stokes approach, that relies on time averaged equations to solve only the mean flow variables, with significant reduction in the computational cost. Figure 2.1 shows the positioning of the different methods on the Kolmogorov cascade and the differences between the signals solved by each method.

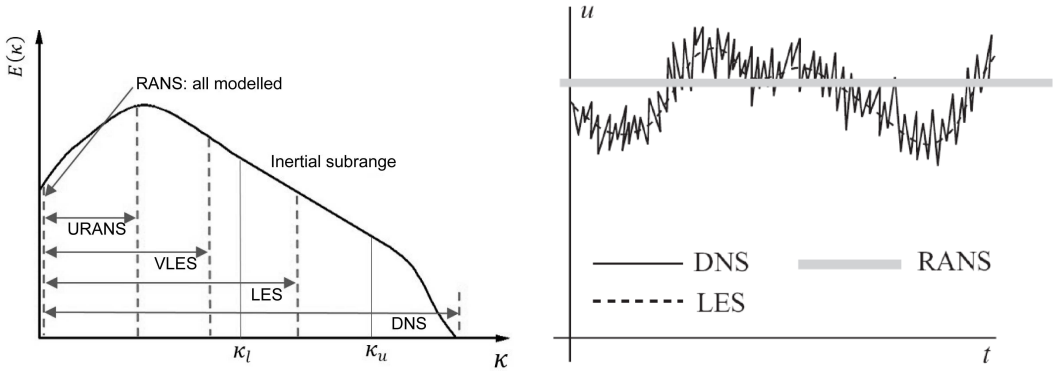


Figure 2.1: Kolmogorov cascade and indication on the modeled scales for different turbulence models (left), with time signals for each model (right) [1]

2.4.1 Direct Numerical Simulation (DNS)

This class of methods relies only on the classical numerical solution of the Navier–Stokes equations down to the level of dissipative scales. This means providing a

fine mesh that is capable to capture these scales, that have to be solved along with all the time scales. In the Kolmogorov theory [23] a measure of the computational power needed is provided using the Reynolds number on a characteristic domain length size L. The total number of grid points for a 3D simulation scales as

$$({}^{L}/_{\Delta x})^3 \sim (Re_L^{3/4})^3 \ , \tag{2.35}$$

likewise, the number of time steps needed is

$${}^{t_L}/_{\Delta t} \sim Re_L^{1/2} \ . \tag{2.36}$$

Therefore the total number of degrees of freedom scales as

$$(Re_L^{3/4})^3 \cdot Re_L^{1/2} = Re_L^{11/4} \ . \tag{2.37}$$

As a quick example the simulation of the flow around a car at 100Km/h would need 10^{20} space-time degrees of freedom. The largest DNS performed today uses $\sim 10^{10}$ grid points on a massively parallel machine, while more generally the common practice on today's machines is around $10^6 - 10^8$ grid points. These limitations are at the base of the continuous developments regarding turbulence models.

2.4.2 Reynolds-Averaged Navier–Stokes (RANS)

In a turbulent flow, pressure and velocity present fluctuations in time. For each generic variable of interest ϕ we can identify the time average $\overline{\phi}$ and the fluctuations ϕ'. Using this notation we can decompose this generic variable as in [24]

$$\phi(\boldsymbol{x}, t) = \overline{\phi}(\boldsymbol{x}) + \phi'(\boldsymbol{x}, t) \ , \tag{2.38}$$

where the time average is computed as

$$\overline{\phi}(\boldsymbol{x}) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \phi(\boldsymbol{x}, t)\ dt \qquad \text{and} \qquad \overline{\phi}' = 0 \ . \tag{2.39}$$

Under the assumption of ergodicity if we use this decomposition on both velocity and pressure in the Navier–Stokes equations, we obtain

$$\begin{cases} \rho(\partial_t \overline{\boldsymbol{v}} + (\overline{\boldsymbol{v}} \cdot \boldsymbol{\nabla})\overline{\boldsymbol{v}}) - \boldsymbol{\nabla}\cdot(2\mu\boldsymbol{\varepsilon}(\boldsymbol{v})) + \boldsymbol{\nabla}\overline{p} + \boldsymbol{\nabla}\cdot(\rho\overline{\boldsymbol{v}' \otimes \boldsymbol{v}'}) = \boldsymbol{f} & \text{in } \Omega \times (0,T) \ , \\ \boldsymbol{\nabla}\cdot\overline{\boldsymbol{v}} = 0 & \text{in } \Omega \times (0,T) \ , \end{cases} \tag{2.40}$$

We can see that in this averaged formulation a new tensor appears, representing the influence of the small scale fluctuation on the mean flow This tensor $\mathcal{R} = -\rho\overline{\boldsymbol{v}' \otimes \boldsymbol{v}'}$ is called Reynolds or turbulent stress tensor, and has to be modeled to close the RANS equations. Under the Boussinesq hypothesis Reynolds stresses are taken into account in the form of an additional diffusion characterized by a turbulent or eddy viscosity μ_t, which models the dissipation of energy due to the correlations of velocity fluctuations with stresses modeled by

$$\mathcal{R} \sim 2\mu_t \boldsymbol{\varepsilon}(\overline{\boldsymbol{v}}) - \frac{2}{3}\rho k \mathbb{I} \ , \tag{2.41}$$

where $k = \frac{1}{2}\mathbf{Tr}(\overline{\boldsymbol{v}' \otimes \boldsymbol{v}'}) = \frac{1}{2}\overline{|\boldsymbol{v}'|^2}$ is the turbulent kinetic energy and $\mathbb{I}$ the identity tensor. Using this hypothesis we obtain the averaged Navier–Stokes momentum equation in the following form

$$\rho(\partial_t \overline{\boldsymbol{v}} + (\overline{\boldsymbol{v}} \cdot \boldsymbol{\nabla})\overline{\boldsymbol{v}}) - \boldsymbol{\nabla}\cdot((\mu + \mu_t)\boldsymbol{\varepsilon}(\overline{\boldsymbol{v}})) + \boldsymbol{\nabla}(\overline{p} + \frac{2}{3}\rho k) = \boldsymbol{f} \ . \tag{2.42}$$

where we can notice that the first term of the Reynolds stress appears as an additional viscosity while the second term is absorbed in the pressure as a turbulent kinetic pressure. To provide a closure to the RANS equations a model is needed to evaluate the turbulent viscosity so that the effective viscosity $\mu_e = \mu + \mu_t$ is computable. The different models available in the literature can be classified according to the number of closure equations to be solved:

- Zero-equation models, use algebraic formulations for μ_t
- One-equation models, use one transport equation for the eddy viscosity or related variables (e.g., Spalart–Allmaras)
- Two-equation models, use two transport equations mostly related to kinetic energy and dissipation (e.g., $k - \epsilon$ or $k - \omega$)

In this work we provide a formulation of the $k-\omega$ SST model, one of the most used two-equation models, that has been implemented and tested in the CIMLIB-CFD library.

2.4.3 $k - \omega$ turbulence model

The $k - \omega$ turbulence model is one of the most used models for RANS equations. It solves two transport equations, for turbulent kinetic energy k, which determines the energy in turbulence, and for the specific turbulent dissipation rate ω, which determines the rate of dissipation per unit turbulent kinetic energy: it is related to k and its dissipation rate ϵ by $\omega = \epsilon/k$.

The standard $k-\omega$ model [25] is more accurate than the popular $k-\epsilon$ in the near wall regions, and can handle flows with moderate adverse pressure gradients, but has problems capturing pressure induced separation. The ω equation also shows a strong sensitivity to the free-stream conditions outside the boundary layer.

Menter's $k-\omega$ SST turbulence model [26], where SST stands for shear stress transport, became popular in the CFD community due to its good performance predicting strong adverse pressure gradient flows, where most of the other turbulence models are not able to provide good results. The SST formulation switches to a $k-\epsilon$ -like behavior in the free-stream, which avoids the problems with free-stream turbulence properties.

The improved near-wall predictions allowed by this turbulent model are crucial to achieve better performance in those simulations involving heat transfer.

2.4.3.1 $k-\omega$ SST model formulation

We give here the complete formulation of the Menter's $k-\omega$ SST turbulence model [26, 27]. The model is built on two transport equations for k and ω. The ω-equation is controlled by blending functions F_1 and F_2 to switch from a $k-\omega$ behavior near the walls, to a $k-\epsilon$ behavior in the free-stream.

$$\begin{cases} \rho\left(\partial_t k + \boldsymbol{v}\cdot\boldsymbol{\nabla} k\right) - \boldsymbol{\nabla}\cdot\left[\left(\mu+\mu_t\sigma_k\right)\boldsymbol{\nabla} k\right] & = P_k - \dfrac{\beta^*\rho}{\nu_t}k^2 \ , \\ \rho\left(\partial_t \omega + \boldsymbol{v}\cdot\boldsymbol{\nabla}\omega\right) - \boldsymbol{\nabla}\cdot\left[\left(\mu+\mu_t\sigma_\omega\right)\boldsymbol{\nabla}\omega\right] & = \dfrac{\gamma P_k}{\nu_t} - \beta\rho\omega^2 \\ & \quad + 2\left(1-F_1\right)\left(\dfrac{\rho\sigma_{\omega 2}}{\omega}\right)\boldsymbol{\nabla} k\cdot\boldsymbol{\nabla}\omega \ , \end{cases} \tag{2.43}$$

where the turbulent kinematic viscosity ν_t is evaluated by relation

$$\nu_t = \frac{k}{\omega}$$

and P_k is the production rate of k

$$P_k = 2\mu_t \|\boldsymbol{\varepsilon}(\overline{\boldsymbol{v}})\|^2 \ , \tag{2.44}$$

which consists of the contribution of the double-contraction of Reynolds shear stresses against the strain-rate tensor.

The turbulent dynamic viscosity μ_t is defined as

$$\mu_t = \frac{\rho a_1 k}{\max\left(a_1\omega, \mathrm{W}F_2\right)} \ , \tag{2.45}$$

where W is the absolute value of the vorticity. Each constant ϕ in the model is computed from the constants ϕ_1 and ϕ_2, using the blending functions F_1, F_2 as follows

$$\phi = F_1\phi_1 + \left(1-F_1\right)\phi_2 \ , \tag{2.46}$$

with the blending functions defined as

$$
\begin{aligned}
F_1 &= \tanh\left(\arg_1^4\right) \ , \\
\arg_1 &= \min\left[\max\left(\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right), \frac{4\rho\sigma_{\omega 2}k}{\mathrm{CD}_{k\omega}d^2}\right] \ , \\
\mathrm{CD}_{kw} &= \max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\boldsymbol{\nabla}k\cdot\boldsymbol{\nabla}\omega, 10^{-20}\right) \ , \\
F_2 &= \tanh\left(\arg_2^2\right) , \\
\arg_2 &= \max\left(2\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right) \ .
\end{aligned}
\tag{2.47}
$$

where d is the distance to the nearest wall.

The closure constants of the model are

$$
\begin{aligned}
&\sigma_{k1} = 0.85, \sigma_{\mathrm{w}1} = 0.65, \beta_1 = 0.075 \ , \\
&\gamma_1 = \beta_1/\beta^* - \sigma_{\omega 1}\kappa^2/\sqrt{\beta^*} \ , \\
&\sigma_{k2} = 1.00, \sigma_{w2} = 0.856, \beta_2 = 0.0828 \ , \\
&\gamma_2 = \beta_2/\beta^* - \sigma_{\omega 2}\kappa^2/\sqrt{\beta^*} \ , \\
&\beta^* = 0.09, \ a_1 = 0.31, \ \kappa = 0.41 \ .
\end{aligned}
\tag{2.48}
$$

2.4.4 Stabilized FEM for solving $k-\omega$ SST

To apply the $k-\omega$ SST to our FE framework we apply the SUPG stabilization, introduced in Subsection 2.2.3, to the transport equations of the model. The idea is to recast the system of equations 2.43 into a convection–diffusion–reaction form applying a backward Euler time discretization.

The system of Equations (2.43) then becomes

$$
\begin{cases}
\dfrac{k^{n+1}-k^n}{\Delta t} + \boldsymbol{v}\cdot\boldsymbol{\nabla}k^{n+1} & -\boldsymbol{\nabla}\cdot\left[\left(\mu+\mu_t^n\sigma_k\right)\boldsymbol{\nabla}k^{n+1}\right] + \dfrac{\beta^*\rho}{\nu_t^n}(k^{n+1})^2 = P_k^n \ , \\
\dfrac{\omega^{n+1}-\omega^n}{\Delta t} + \boldsymbol{v}\cdot\boldsymbol{\nabla}\omega^{n+1} & -\boldsymbol{\nabla}\cdot\left[\left(\mu+\mu_t^n\sigma_\omega\right)\boldsymbol{\nabla}\omega^{n+1}\right] + \beta\rho(\omega^{n+1})^2 = \dfrac{\gamma P_k^n}{\nu_t^n} \\
& -2\left(1-F_1^n\right)\left(\dfrac{\rho\sigma_{\omega 2}}{\omega^n}\right)\boldsymbol{\nabla}k^n\cdot\boldsymbol{\nabla}\omega^n \ .
\end{cases}
\tag{2.49}
$$

Provided that the square term is linearized we can identify the terms of convection, diffusion, reaction, and source. The system is then discretized in space using the SUPG method. The Galerkin formulation is obtained by multiplying the two equations by appropriate test functions, $\varphi_k+\tau_k\boldsymbol{v}\cdot\boldsymbol{\nabla}\varphi_k$, and $\varphi_\omega+\tau_\omega\boldsymbol{v}\cdot\boldsymbol{\nabla}\varphi_\omega$. We then apply the divergence theorem to the diffusion terms and integrate by parts, following the

lines in [28] on the use of stabilization methods for transient convection–diffusion–reaction equations. The k equation becomes

$$\left(\frac{k^{n+1}-k^{n}}{\Delta t},\varphi_k\right)+\left(\boldsymbol{v}^{n+1}\cdot\boldsymbol{\nabla}k^{n+1},\varphi_k\right)$$
$$+\left(\left[\mu+\mu_t^n\sigma_k\right]\boldsymbol{\nabla}k^{n+1},\boldsymbol{\nabla}\varphi_k\right)+\left(\frac{\beta^*\rho k^n}{\nu_t^n}k^{n+1},\varphi_k\right)$$
$$-\left(P_k^n,\varphi_k\right)+\sum_K\left(\mathcal{R}(k^{n+1}),\tau_k^{n+1}\left(\boldsymbol{v}^{n+1}\cdot\boldsymbol{\nabla}\varphi_k\right)\right)_K=0\quad,\quad(2.50)$$

and the ω equation becomes

$$\left(\frac{\omega^{n+1}-\omega^{n}}{\Delta t},\varphi_\omega\right)+\left(\boldsymbol{v}^{n+1}\cdot\boldsymbol{\nabla}\omega^{n+1},\varphi_\omega\right)$$
$$-\left(\left[\mu+\mu_t^n\sigma_\omega\right]\boldsymbol{\nabla}\omega^{n+1},\boldsymbol{\nabla}\varphi_\omega\right)+\left(\beta\rho\omega^n\omega^{n+1},\varphi_\omega\right)$$
$$-\left(\frac{\gamma P_k^n}{\nu_t^n}-2\left(1-F_1^n\right)\left(\frac{\rho\sigma_{\omega2}}{\omega^n}\right)\boldsymbol{\nabla}k^n\boldsymbol{\nabla}\omega^n,\varphi_\omega\right)$$
$$+\sum_K\left(\mathcal{R}(\omega^{n+1}),\tau_\omega^{n+1}\left(\boldsymbol{v}^{n+1}\cdot\boldsymbol{\nabla}\varphi_\omega\right)\right)_K=0\quad,\quad(2.51)$$

where $\mathcal{R}(.)$ is the finite element residual of each equation in System (2.49). The stabilization parameters, $\tau_{(.)}$, is computed within each element as

$$\tau_{(.)}|_K=\left(\frac{c_2}{h}\left\|\alpha_c\right\|_K+\frac{c_1}{h_K^2}\alpha_d+\alpha_r\right)^{-1}\quad(2.52)$$

where α_c,α_d, and α_r are, respectively, the convection, diffusion, and reaction coefficients for the k- and ω-equations, h_K is the element size, $\left\|\alpha_c\right\|_K$ a characteristic norm of the convection term, and $c_1=4,c_2=2$ for linear elements.

2.5 Conjugate heat transfer

The coupling between heat transfer and turbulent fluid flow problems is a strong subject of research in the past and nowadays still. This interest is driven by the numerous industrial application that involve this type of physics, specially in the field of material processing. Engineers are strongly interested, e.g., in the temperature distribution inside a treated piece during all the process, to determine the metallurgical properties at the end of the treatment. Having all the thermal information at

hand allows an optimization of the design of the tools and of the process itself, with possibly a reduction in the energy consumption and pollutant emission.

To perform this task, the CFD tool needs to couple efficiently the two types of governing equations presented above, considering a framework that allows a multi-domain simulation. In this work we intend to employ the Immersed Volume method as presented in [29]. The method consists in using a single global mesh to discretize the solid and fluid sub-domains and solve one set of equations with different thermo-mechanical properties. Using this approach, we avoid the challenging task of determining heat transfer coefficients at the level of the fluid-solid interface. Simulating the conjugate heat transfer and flow dynamics inside the domain requires the simultaneous resolution of the Navier–Stokes and the heat transfer equations. In convection-dominated regimes, the equations are stabilized using the aforementioned SUPG and VMS approaches (Sections 2.2.3 and 2.3.3 respectively). The fluid dynamics and heat transfers are modeled by the CDR equations and RANS equations with the $k-\omega$ closure model. The Boussinesq approximation is used to take into account the buoyancy forces on the fluid motion. The coupled system to solve follows

$$\begin{cases} \boldsymbol{\nabla}\cdot\boldsymbol{v} = 0 & \text{in } \Omega\ , \\ \rho\left(\partial_t\boldsymbol{v} + (\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{v}\right) - \boldsymbol{\nabla}\cdot\left(2\mu_e\boldsymbol{\varepsilon}(\boldsymbol{v}) - p\mathbb{I}\right) = \rho_0\beta\left(T-T_0\right)\mathbf{g} & \text{in } \Omega\ , \\ \rho C_p\left(\partial_t T + \boldsymbol{v}\cdot\boldsymbol{\nabla}T\right) - \boldsymbol{\nabla}\cdot\left(\lambda_e\boldsymbol{\nabla}T\right) = f - \boldsymbol{\nabla}\cdot\mathbf{q_r} & \text{in } \Omega\ , \end{cases} \tag{2.53}$$

where $\boldsymbol{v}$ denotes the velocity field, p the effective pressure, T the temperature, $\boldsymbol{\varepsilon}(\boldsymbol{v})$ the strain-rate tensor, ρ the density and μ_e the effective dynamic viscosity; ρ_0 and T_0 are reference density and temperature, β the thermal expansion coefficient and $\mathbf{g}$ the gravitational acceleration. In the energy conservation equation we consider the constant pressure specific heat capacity C_p, the effective thermal conductivity λ_e, a volume source term f and the heat radiative flux $\mathbf{q_r}$. The effective viscosity μ_e and effective thermal conductivity λ_e terms which are computed by

$$\mu_e = \mu + \mu_t \quad \text{and} \quad \lambda_e = \lambda + \frac{C_p\mu_t}{\mathrm{Pr}_t}\ , \tag{2.54}$$

with $\mathrm{Pr}_t = 0.85$ being the turbulent Prandtl number. The turbulent viscosity μ_t is obtained from the closure equations of the turbulent model. It is important to mention that we omitted the over-lined notation used previously for the averaged variables ($\boldsymbol{v}, p$ and T) for sake of simplicity. When using the RANS method, all the variables stand for their averaged counterpart.

2.6 Numerical validation

In order to validate the proposed numerical framework we present in this section several benchmark test cases. First we show the performances of the VMS coupled

with CDR to model the heat transfer in a natural convection test. Then we validate the $k-\omega$ method proposed for RANS equations, on a turbulent flow.

2.6.1 Natural convection in a square cavity

We present here the common benchmark with the laminar flow in a two-dimensional square cavity, analyzed in [30, 31]. The velocity and the temperature equations are coupled due to the buoyancy force and solved. Consequently, the flow inside the enclosure is driven by the temperature differences. Figure 2.2(a) shows the setup of the problem with the right wall kept at a constant cold temperature, whereas the left wall is at a constant high temperature. Other two walls are maintained at adiabatic condition. The gravitational acceleration is taken parallel to the isothermal walls. In this test we describe the physical properties using the Grashof number

$$Gr = \frac{\rho^2 g\beta \left(T_h - T_c\right) L^3}{\mu^2} \ , \tag{2.55}$$

where ρ is the density of the fluid, g is the gravity constant, β is the coefficient of thermal expansion of the fluid, $\Delta T = T_h - T_c$ is the temperature difference between the hot and cold walls in the figure separated by width L, and μ the dynamic viscosity. This nondimensional number describes the ratio of the buoyancy to viscous forces acting on a fluid. We assume that the fluid properties are to be constant, except for the density in the buoyancy term. Under this assumption the Boussinesq approximation states that density differences are sufficiently small to be neglected, except where they appear in terms multiplied by g, the acceleration due to gravity. The force term in the momentum equation then becomes

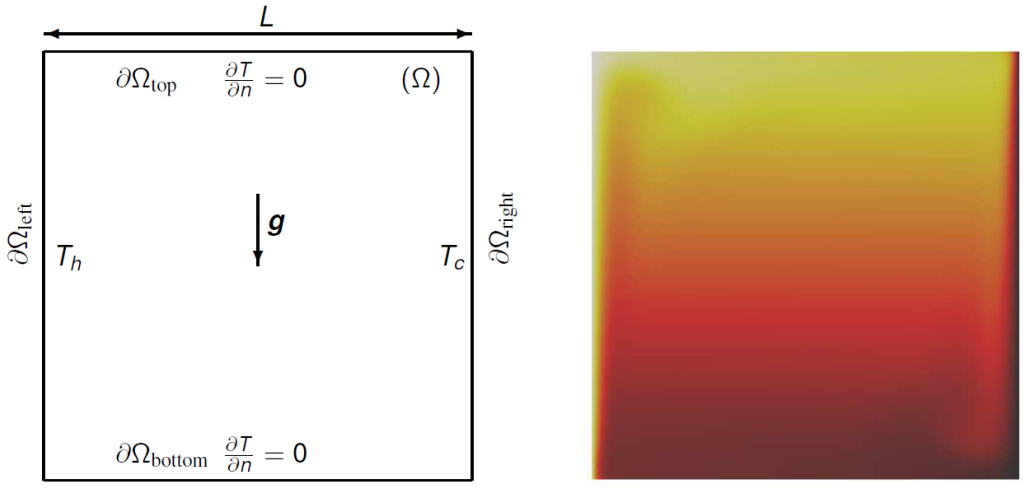
$$\left(\rho - \rho_0\right) \mathbf{g} \approx -\rho_0 \beta \left(T - T_0\right) \mathbf{g} \ . \tag{2.56}$$

where ρ_0 is the constant density of the flow, β is again the thermal expansion coefficient, and T_0 is the operating temperature.

The contours of the temperature field for $Gr = 10^7$ are shown in Figure 2.2(b). In Figure 2.3(b) we plot the nondimensionalized velocity profile in the boundary layer at $y = 0.5$, for three different Gr values. The analysis performed in [32] predicts that the velocity profile in the boundary layer should scale with the power $1/4$ of the Gr number. We show in Figure 2.3(b) that the curves plotted for three different Gr numbers follow this scaling low exactly as predicted.

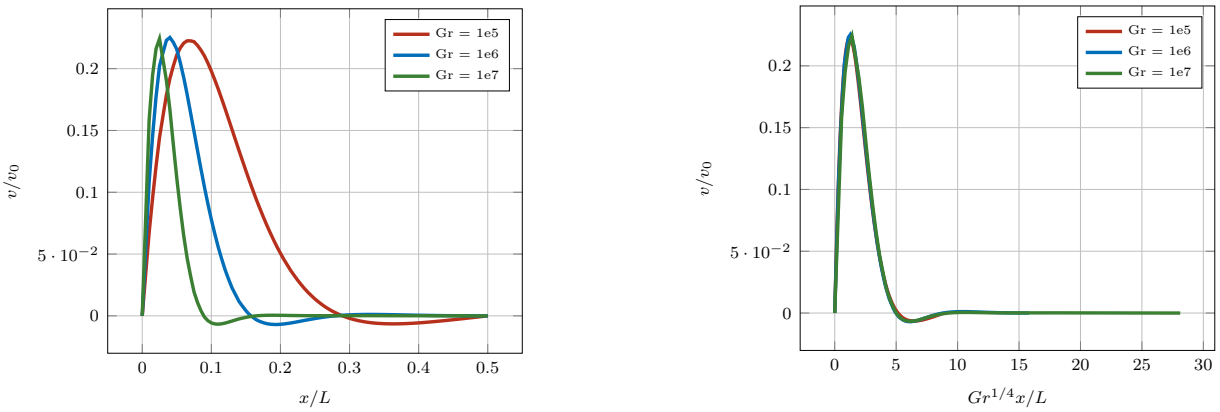
2.6.2 Jet impingement

Owing to the highly localized mass, momentum, and heat transfer, jet impingement is widely used in various fields such as cooling or heating of surfaces. However,



(a) Problem setup (b) Temperature distribution

Figure 2.2: Natural convection in a square cavity



(a) Velocity profile (b) Velocity profile scaled

Figure 2.3: Natural convection, velocity profiles

despite the apparent geometric simplicity, the flow structure of jet impingement is complex, as it involves strong adverse pressure gradient, separation, vortex formation and breakdown. Given the high Reynold number of the flow a RANS turbulence model is used in the form of the $k-\omega$ SST variant.

The computational domain of the studied case [33] is depicted in Figure 2.4. Because of the symmetry of the configuration only half of the domain is simulated, allowing to reduce the computational cost,. The inlet width is $B = 0.04m$, the impinging plate width is $4m$, and consequently the confined plate width is $3.96m$.

Boundary	$\boldsymbol{u}$	p	k	ω	T
Inlet	$\boldsymbol{u} = (0, -\mu Re/B)^T$	$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0$	$k = \lvert I\boldsymbol{u}\rvert^2$	$\omega = \dfrac{4\sqrt{k}}{27 l_c \beta^*}$	$T = 300$
Outlet	$\boldsymbol{\nabla u} \cdot \boldsymbol{n} = 0$	$p = 0$	$k = 0$	$\omega = 0$	$T = 300$
Impinging	$\boldsymbol{u} = \boldsymbol{0}$	$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0$	$k = 0$	wall law	$T = 310$
Confined	$\boldsymbol{u} = \boldsymbol{0}$	$\boldsymbol{\nabla} p \cdot \boldsymbol{n} = 0$	$k = 0$	wall law	$T = 300$

Table 2.1: Boundary conditions for the jet impingement case

For this case, the ratio is chosen as $H/B = 4$ so that the impinging distance is $0.16m$.

Figure 2.4: Setup of the jet impingement case

Particular attention is given to the mesh generation procedure given the sensitivity to the boundary layer. The resolution in the vicinity of the plates should satisfy a condition on the y^+ value to ensure that it does not impact the behavior of the flow in the near-wall region; given the references for this case we work with a finite element mesh consisting of 368×126 elements.

The Reynolds number is $Re = 20000$, the Prandtl number is $Pr = 0.72$, and for the inlet the turbulent intensity is set to $I = 0.01$. The characteristic turbulent length scale is $l_c = 0.015B = 0.0006m$, the fluid density is $\rho = 1.1716kg/m^3$, the dynamic viscosity is $\mu = 1.835 \cdot 10^-5 kg/m \cdot s$, the thermal conductivity is $\kappa = 0.0263W/m \cdot K$, and the specific heat is $C_p = 1005.5W/kg \cdot K$. The density is considered constant and since the maximum temperature difference is only $\Delta T = 10K$, the Boussinesq approximation was applied. Finally the boundary conditions listed in Table 2.1 were enforced.

The obtained final velocity magnitude and turbulent kinetic energy fields are presented on Figures 2.5 and 2.6 while a comparison of the pressure and velocity profiles is provided in Figures 2.7(a) and 2.7(b).

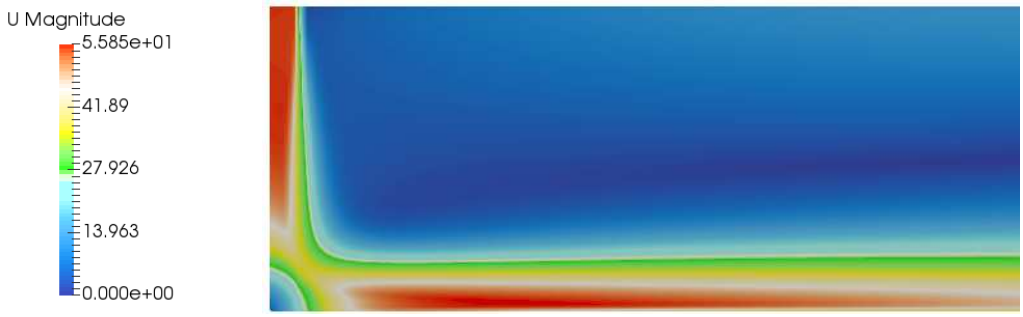


Figure 2.5: Velocity magnitude for the jet impingement case

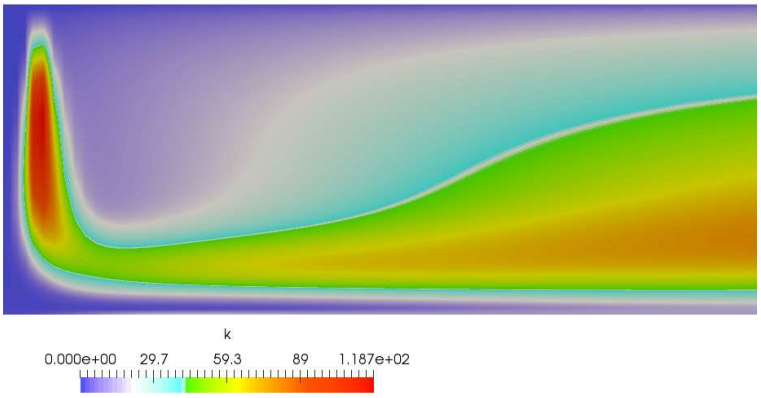
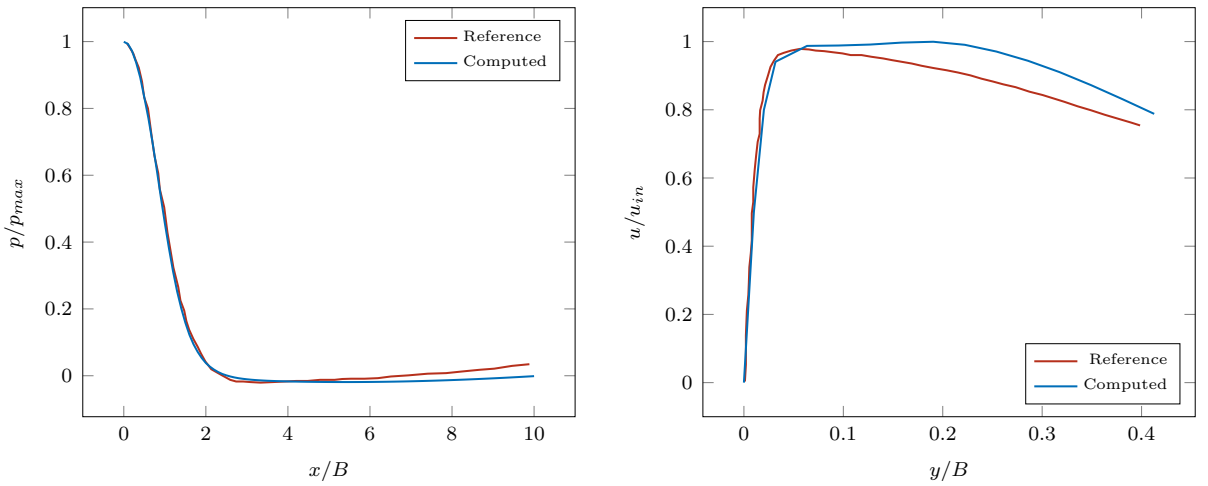


Figure 2.6: Turbulent kinetic energy for the jet impingement case

2.7 Conclusions

In this chapter we presented the main framework for the numerical resolution of fluid flows and conjugate heat transfer problems. A finite element discretization of the computational domain is adopted for its accuracy and efficiency with the complex geometries that are typical of industrial simulations. However the standard Galerkin formulation is affected by numerical oscillations in convection or diffusion dominated problems especially in the presence of sharp gradients. To overcome this limitation we chose the Streamline Upwind Petrov–Galerkin (SUPG) method applied with convection-dominated convection–diffusion–reaction problems. This approach is efficient to eliminate the instabilities adding a numerical diffusion in the



(a) Normalized pressure at $y = 0$

(b) Velocity normalized by inlet at $x/B = 2$

Figure 2.7: Simulation of turbulent slot jet impingement heat transfer, comparison with reference results from [33]

convective direction. To solve the fluid flow problem we use a variational multiscale approach to solve the incompressible Navier–Stokes equations. This stabilization method is proven to be efficient in handling convection-dominated problems and deal with the inf-sup stability condition. When dealing with turbulent flows, we introduced the RANS equations with turbulence models, that provide the averaged fluid flow solution without solving the turbulence small scales. All these ingredients will be used in the following with the aim of solving complex industrial heat transfer problems. However to solve all these equations for industrial applications, one needs to rely on iterative procedures. The aim of this work is to provide an adaptive control for the iterative solvers, to make the solution procedure faster and more efficient.

2.8 Bibliography

[1] E. Hachem, Stabilized finite element method for heat transfer and turbulent flows inside industrial furnaces, Ph.D. thesis (2009). v, 11, 22

[2] E. Hachem, G. Jannoun, J. Veysset, T. Coupez, On the stabilized finite element method for steady convection-dominated problems with anisotropic mesh adaptation, Applied Mathematics and Computation 232 (2014) 581 – 594. 14

[3] E. Hachem, B. Rivaux, T. Kloczko, H. Digonnet, T. Coupez, Stabilized finite element method for incompressible flows with high reynolds number, Journal of Computational Physics 229 (23) (2010) 8643–8665. 15

[4] E. Hachem, G. Jannoun, J. Veysset, M. Henri, R. Pierrot, I. Poitrault, E. Massoni, T. Coupez, Modeling of heat transfer and turbulent flows inside industrial furnaces, Simulation Modelling Practice and Theory 30 (2013) 35–53. 11

[5] A. N. Brooks, T. J. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering 32 (1-3) (1982) 199–259. 11

[6] T. J. Hughes, L. P. Franca, M. Balestra, A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations, Computer Methods in Applied Mechanics and Engineering 59 (1) (1986) 85–99. 11

[7] G. Scovazzi, A discourse on Galilean invariance, SUPG stabilization, and the variational multiscale framework, Computer Methods in Applied Mechanics and Engineering 196 (4-6) (2007) 1108–1132. 11

[8] H. Nguyen, M. Gunzburger, L. Ju, J. Burkardt, Adaptive anisotropic meshing for steady convection-dominated problems, Computer Methods in Applied Mechanics and Engineering 198 (37-40) (2009) 2964–2981. 12

[9] I. Babuška, The finite element method with lagrangian multipliers, Numerische Mathematik 20 (3) (1973) 179–192. 15

[10] F. Brezzi, On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers, Publications mathématiques et informatique de Rennes (S4) (1974) 1–26. 15

[11] D. N. Arnold, F. Brezzi, M. Fortin, A stable finite element for the stokes equations, Calcolo 21 (4) (1984) 337–344. 15

[12] L. P. Franca, A. Nesliturk, On a two-level finite element method for the incompressible navier–stokes equations, International Journal for Numerical Methods in Engineering 52 (4) (2001) 433–453. 15

[13] F. Brezzi, L. Franca, T. J. Hughes, A. Russo, Stabilization techniques and subgrid scales capturing, Consiglio Nazionale delle Ricerche. Istituto di Analisi Numerica, 1996. 15

[14] R. Codina, Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods, Computer Methods in Applied Mechanics and Engineering 190 (13-14) (2000) 1579–1599. 15, 20, 21

[15] R. Codina, Pressure stability in fractional step finite element methods for incompressible flows, Journal of Computational Physics 170 (1) (2001) 112–140.

[16] R. Codina, J. Principe, Dynamic subscales in the finite element approximation of thermally coupled incompressible flows, International Journal for Numerical Methods in Fluids 54 (6-8) (2007) 707–730. 15, 20

[17] T. J. Hughes, Multiscale phenomena: Green's functions, the dirichlet-to-neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Computer Methods in Applied Mechanics and Engineering 127 (1-4) (1995) 387–401. 15

[18] T. J. Hughes, G. R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method—a paradigm for computational mechanics, Computer Methods in Applied Mechanics and Engineering 166 (1-2) (1998) 3–24.

[19] T. J. Hughes, A. A. Oberai, L. Mazzei, Large eddy simulation of turbulent channel flows by the variational multiscale method, Physics of fluids 13 (6) (2001) 1784–1799. 15

[20] T. Dubois, F. Jauberteau, R. Temam, Dynamic multilevel methods and the numerical simulation of turbulence, Cambridge University Press, 1999. 19

[21] T. Coupez, Stable-stabilized finite element for 3d forming calculation. cemef, Tech. rep., ENSMP, internal report (1996). 19

[22] T. E. Tezduyar, Y. Osawa, Finite element stabilization parameters computed from element matrices and vectors, Computer Methods in Applied Mechanics and Engineering 190 (3-4) (2000) 411–430. 21

[23] A. N. Kolmogorov, The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 434 (1890) (1991) 9–13. 23

[24] L. Gaston, Simulation numérique par éléments finis bidimensionnels du remplissage de moules de fonderie et étude expérimentale sur maquette hydraulique, Ph.D. thesis, ENSMP (1997). 23

[25] D. C. Wilcox, Reassessment of the scale-determining equation for advanced turbulence models, AIAA journal 26 (11) (1988) 1299–1310. 25

[26] F. R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, AIAA journal 32 (8) (1994) 1598–1605. 25

[27] F. R. Menter, M. Kuntz, R. Langtry, Ten years of industrial experience with the SST turbulence model, Turbulence, heat and mass transfer 4 (1) (2003) 625–632. 25

[28] S. Badia, R. Codina, Analysis of a stabilized finite element approximation of the transient convection-diffusion equation using an ALE framework, SIAM Journal on Numerical Analysis 44 (5) (2006) 2159–2197. 27

[29] R. Valette, T. Coupez, C. David, B. Vergnes, A direct 3d numerical simulation code for extrusion and mixing processes, International Polymer Processing 24 (2) (2009) 141–147. 28

[30] G. de Vahl Davis, Natural convection of air in a square cavity: a bench mark numerical solution, International Journal for Numerical Methods in Fluids 3 (3) (1983) 249–264. 29

[31] A. Yücel, S. Acharya, M. Williams, Natural convection and radiation in a square enclosure, Numerical Heat Transfer 15 (2) (1989) 261–278. 29

[32] A. Bejan, Convection heat transfer, 3rd Edition, John Wiley & Sons, Hoboken, New Jersey, 2013. 29

[33] M. A. R. Sharif, K. K. Mothe, Evaluation of turbulence models in the prediction of heat transfer due to slot jet impingement on plane and concave surfaces, Numerical Heat Transfer, Part B: Fundamentals 55 (4) (2009) 273–294. v, 30, 33

Chapter 3

Error estimation for anisotropic mesh adaptation

Contents

Résumé *Dans ce chapitre, nous présentons brièvement la méthode d'adaptation de maillage anisotrope basée sur la métrique utilisée pour fournir un maillage optimal, capable de capturer des gradients abrupts dans les domaines d'intérêt. Nous décrivons les principes de l'approche basée sur la métrique, pour construire un tenseur métrique dans l'espace de Riemann. Le résultat de la procédure est un ensemble de facteurs d'étirement qui sont utilisés pour adapter le maillage, résultant en un maillage anisotrope bien adapté au domaine d'intérêt. La procédure d'adaptation est pilotée par un estimateur d'erreur. Nous proposons deux estimateurs d'erreur a posteriori, l'un basé sur le hessien récupéré de la solution, le second s'appuyant sur les gradients récupérés sur le bord des éléments. Les deux estimateurs sont généraux et indépendants de l'EDP, et peuvent être appliqués à n'importe quel domaine d'intérêt. Dans le chapitre suivant, nous proposons d'utiliser le même estimateur, déjà calculé pour l'adaptation du maillage, pour piloter la convergence de la solution itérative.*

3.1 Introduction

Computational Fluid Dynamics simulations and thermal analysis are nowadays becoming more and more accessible and are being integrated in most of the engineering design and optimization processes. Nevertheless the modeled phenomena remain extremely complex and those simulations require considerable computational resources. A traditional uniform-mesh approach often fails to provide good solutions with a reasonable number of elements, especially in the presence of multi-scale features or steep gradients. Mesh adaptation techniques answer to this limitation with a local refinement in the regions where higher resolution is needed. However, to capture directional features that characterize the solution fields, the elements of the mesh need to be highly stretched and well oriented, following these features. Anisotropic adapted meshes provide more accurate solutions, reducing the required computational resources. In this work we use a metric-based mesh adaptation technique to obtain anisotropic adapted meshes based on an error estimator. We propose two different error estimators, one based on the recovered Hessian of the solution and one computed on the edges of the element.

In Section 3.2 we outline different mesh adaptation techniques, with a focus on the metric-based anisotropic mesh adaptation. We continue in Section 3.3 we present the a-posteriori error estimators proposed and the related gradient recovery technique. We conclude in Section 3.4 with some numerical examples to show the efficiency of the mesh adaptation procedure.

3.2 Mesh adaptation and error estimates

Introduced in the late 70s, the concept of mesh adaptation has been the focus of intensive studies to have more accurate and faster numerical simulations. Error estimators were used as indicators for the adaptation procedure, to determine the mesh sizes accordingly in the different regions [1–4]. These methods, however, require still a considerable number of elements in presence of steep gradients in the solution fields. When the solution is characterized by the presence of directional features such as boundary layers, the mesh should be able do adapt to this anisotropy. Anisotropic mesh adaptation provides control on the size shape and orientation of the elements, to better represent the solution features with a reduced computational cost. The first studies on anisotropic mesh adaptation were based on a moving front technique [5, 6]. In the 90s a new approach was introduced, generating highly stretched elements in a locally mapped space with metric tensors [7–9]. This method however, can often lead to the creation of poor quality elements that would spoil the accuracy of the solution, and several methods have been proposed to provide quality improvements for the mesh [10, 11]. Local-remeshing is proposed as a possible solution to overcome this problem [12–14]. In the following we present the outlines of the metric-based approach following the studies developed by the CFL research group at CEMEF.

3.2.1 Metric-based mesh adaptation

Non Euclidean metrics are often used to generate unstructured anisotropic meshes [15]. The main idea of metric-based mesh adaptation, is to generate a unit mesh in a prescribed Riemannian metric space, such that this mesh can be created by any unstructured uniform mesh generator. Then, this mesh is converted in an unstructured anisotropic mesh by the inverse application in the physical space. A discrete metric $\mathcal{M}$ consists of a family of symmetric positive definite linear forms, each of them represented by a real matrix M in $\mathbb{R}^{d\times d}$ which can therefore be diagonalized, so that

$$\mathrm{M} = \mathrm{R}^T \Lambda \mathrm{R} \ , \tag{3.1}$$

where R is an orthogonal matrix the lines of which are composed of eigenvectors $(\mathbf{r}_i)_{i=1..d}$ while Λ is the diagonal matrix composed of eigenvalues $(\lambda_i)_{i=1..d}$, which are strictly positive. Then we can define the inner-product of two vectors in $\mathbb{R}^d$ with respect to the metric M :

$$(\mathbf{u}, \mathbf{v})_{\mathrm{M}} = (\mathbf{u}, \mathrm{M}\mathbf{v}) = \mathbf{u}^T \, \mathrm{M} \, \mathbf{v} \in \mathbb{R} \ . \tag{3.2}$$

Hence the distance between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined by the M-conjugate inner-product

$$\mathrm{d}_{\mathrm{M}} \left(\mathbf{x}_i, \mathbf{x}_j \right) = \left(\mathbf{x}_{ij} \, \mathrm{M} \, \mathbf{x}_{ij} \right)^{\frac{1}{2}} \tag{3.3}$$

with $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$, and the volume of an element K is $|K|_{\mathrm{M}} = \sqrt{\det(\mathrm{M})}|K|_{\mathbb{I}}$, with $\mathbb{I}$ the identity matrix defining the canonical metric. We can then define the unit ball $\mathcal{B}_{\mathrm{M}}$ by

$$\mathcal{B}_{\mathrm{M}} = \left\{ \mathbf{x} \in \mathbb{R}^d \mid \mathbf{x}^T \mathrm{M} \mathbf{x} \leq 1 \right\} . \tag{3.4}$$

The unit ball is thus delimited by an ellipsoid with axis the eigenvectors $\mathbf{r}_i$ and sizes $h_i = (\lambda_i)^{-1/2}$ along these directions, as shown in Figure 3.1. Consequently, we can deduce a linear mapping between the unit ball in the canonical Euclidean space and the metric space, represented in Figure 3.1.

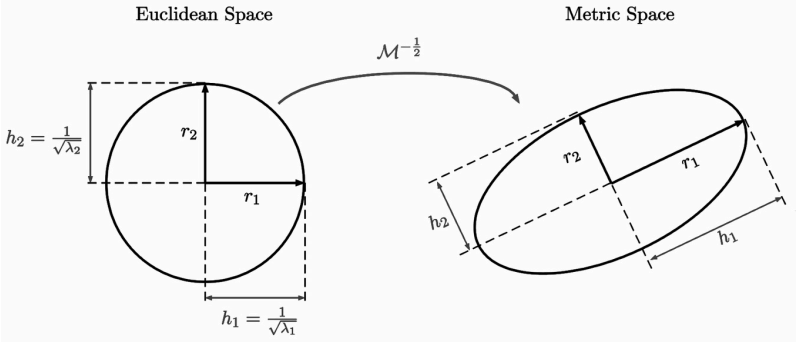


Figure 3.1: Unit ball and linear mapping between the metric space and the canonical Euclidean space

3.2.2 Error estimates

The goal of any anisotropic mesh adaptation technique is to provide a discretization that minimizes the discrepancy between the approximated and the exact solution. With a reliable approximation error, this can be done increasing the mesh density in the regions where this error is estimated to be higher. This error estimator is then used to drive the creation of the metric tensor that modifies the mesh sizes and directions.

There are two main classes of error estimator: a-priori and a-posteriori. A-priori estimators rely only on information available at the definition of the problem, such as the nature of the problem itself or quantities related to the desired mesh, and on the exact solution. However, for CFD applications a-priori error estimates are frequently unreliable, especially in presence of singularities, and are not able to provide directional information. A-posteriori estimators, on the other side, rely on the approximated solution of the problem. Different approaches can be followed to provide this type of estimators. Residual-based estimators [16, 17] are derived from the residual of the discretized problem, and are therefore problem dependent. Hierarchical estimators [18, 19] use a higher order approximation of the exact solution to

evaluate the error of the actual (lower order) computed solution. Goal-oriented estimators [20–22] take advantage from the solution of the adjoint problem to optimize a given functional of interest, that is imposed as a source term of the adjoint. The provided mesh would thus be only adapted to capture the features of given functional. The last family of estimators is based on gradient recovery techniques [23–25]. A limitation of this type of estimator is frequently related to the computational cost of the recovery technique used. In the following we present two estimators of this family. The first provides an edge-based estimate [26] using a simple gradient recovery technique. The second relies on the recovered Hessian of the solution to drive an explicit optimization problem [27–29]. Both approaches use the equi-distribution principle to provide a set of stretching factors associated to the edges of the actual mesh, for a given number of vertices.

It is worth mentioning that a combination of the above technique is possible. Residual-based estimators, e.g., can be used in combination with other type of estimators, often as a scaling contribution. In [30] Bazile proposes a combination with a novel approach using an estimator based on the modeling of the VMS subscales (from eq. 2.25). Another possible combination found in [31, 32] uses a coupling with a directional contribution from an Hessian based estimator.

3.2.3 Global optimization

The techniques that we outline in this section is based on the error equi-distribution principle. A global optimization problem is formulated under the constraint of a fixed number of vertices in the mesh, to provide a set of stretching factors associated with the edges of the domain. The length distribution tensor is modified accordingly to take into account these stretching factors resulting in the new anisotropic adapted mesh.

Let $\mathcal{T}_h$ be a finite element admissible mesh of the domain $\Omega \cap \mathbb{R}^d$ such that $\Omega = \bigcup_{K \in \mathcal{T}_h} K$, being K a simplex (i.e., a triangle in 2D, a tetrahedron in 3D), such that $K_i \cap K_j = \emptyset$ for any pair of cells (K_i, K_j) of $\mathcal{T}_h$. We define the set of vertex coordinates in the mesh by

$$\mathbf{x} = \{\mathbf{x}_i \in \mathbb{R}^d, i = 1, ..., N\} \ , \tag{3.5}$$

where each vertex is identified by the unique index i in the mesh topology. Additionally we consider the star-topology of $\mathcal{T}_h$ defined by the family of set of vertices $\Gamma(i)$ connected to a vertex of index i so that

$$\Gamma(i) = \{j : \exists K \in \mathcal{T}_h, i, j \text{ are vertices of } K\} \ , \tag{3.6}$$

in other words the pair (i, j) defines an edge of $\mathcal{T}_h$.

In the following we consider $\mathcal{V}_h \subset \mathcal{V} = \mathrm{H}^1(\Omega)$ the linear Lagrange finite element approximation space

$$\mathcal{V}_h = \{w_h \in \mathrm{C}^0(\overline{\Omega}) : w_h|_K \in \mathbb{P}_1(K), K \in \mathcal{T}_h\} \ , \tag{3.7}$$

and with the Lagrange interpolation operator from $\mathcal{V}$ to $\mathcal{V}_h$ defined as

$$\begin{array}{rccc} \Pi_h : & \mathcal{V} & \to & \mathcal{V}_h \\ & u & \mapsto & \displaystyle\sum_{i=1}^{N} \mathrm{u}^i \phi_i \end{array} \tag{3.8}$$

where $\{\phi_i\}_{i=1,\cdots,N}$ is a nodal basis of $\mathcal{V}_h$, and real coefficients $\{\mathrm{u}^i = u(\mathbf{x}_i)\}_{i=1,\cdots,N}$ are the degrees of freedom; therefore we identify u^i as the nodal value of u at the vertex of coordinates $\mathbf{x}_i$.

We recall that the goal of the adaptation procedure is to find a discretization that minimizes the approximation error in a given norm $\|u - u_h\|$ for a target cost. Estimates of the approximation error are said *a priori* as they hold for any solution u of the continuous problem and are useful to determine the convergence properties of the numerical method. However they are not computable and therefore cannot be used to estimate the quality of a given approximate solution. Owing to *Cea's Lemma* the approximation error is uniformly bounded by the interpolation error [33, 34]:

$$\|u - u_h\|_{\mathrm{H}^1} \leq C\|u - \Pi_h u\|_{\mathrm{H}^1} \ , \tag{3.9}$$

where the constant C depends on the problem (e.g, it involves the continuity and the coercivity constants).

Therefore the mesh adaptation procedure can be formulated as a global optimization problem to minimize the interpolation error in the given norm, with a constraint on the number of degrees of freedom. In the frame of metric-based mesh adaptation, a discrete metric field $\mathcal{M}$, consisting of a family of symmetric positive definite linear forms defined at the vertices of $\mathcal{T}_h$ is computed so that

$$\mathcal{M} = \underset{u}{\operatorname{argmin}} \ \|u - \Pi_h u\| \tag{3.10}$$

The adaptation algorithm relies therefore on a quality measure of the computed solution which estimates the approximation error in order to build a computable minimization Problem (3.10), it is called an *a posteriori* error estimator.

3.3 A posteriori error estimation

Among the possible choices outlined in Section 3.2.2, the two approaches that we propose in the following both rely on the gradients of the solution to drive the adaptation.

3.3.1 Hessian recovery techniques

In a P_1 finite element framework the gradients of the solution are constant on the element. To be able use them to control the approximation error, we need to find a suitable recovery technique to overcome this problem. The most common methods found in the literature aim to recover the Hessian of the solution at the mesh nodes, to use it in the mesh adaptation procedure.

One well-known class are projection methods, like the Zienkiewicz–Zhu approach [24], used on P_1 finite elements. In order to recover the gradient at the nodes, that is discontinuous from element to element, a patch of elements sharing a node $\mathbf{x}_i$ is built. On this patch a least square linear fitting function is defined, considering the gradient values in the element's center of mass. The Hessian is obtained applying this reconstruction strategy twice recursively.

$$\mathbb{H}_R(u_h)(\mathbf{x}_i) = \Pi_{ZZ}\left(\boldsymbol{\nabla}\left(\Pi_{ZZ}\left(\boldsymbol{\nabla} u_h\right)\right)\right) i \ , \tag{3.11}$$

with Π_{ZZ} the Zienkiewicz–Zhu projector on the P_1 finite element space. The two main limits of this class are the numerical diffusion introduced by the recursive averaging process, and the incapability of removing high frequency errors due to mesh non-uniformity.

Another possible approach is to use variational methods that rely on the gradient to be constant elementwise, for a P_1 scalar field. The Hessian is recovered using a weak formulation combined with the Green formula, considering a higher order approximation of the scalar field, see e.g. [35].

The last class proposed is based on a least square approach to recover the Hessian at the nodes. Considering a node $\mathbf{x}_j$ on a patch of elements sharing the node $\mathbf{x}_i$, the Hessian can be written with a Taylor expansion truncated at the second order

$$\frac{1}{2}\mathbf{x}_{ij}\mathbb{H}_R(u_h)(\mathbf{x}_i)\mathbf{x}_{ij} = u_h(\mathbf{x}_j) - u_h(\mathbf{x}_i) - \mathbf{x}_{ij}\boldsymbol{\nabla} u_h(\mathbf{x}_i) \ . \tag{3.12}$$

This equation can be reformulated as a system that can be solved using a least square approximation

$$AH = B \ , \tag{3.13}$$

where the components of the Hessian are in the vector H. The fact that system has to be solved at each remeshing iteration on each node, makes the complexity of this approach to be problem dependent.

3.3.2 Hessian based estimator and mesh adaptation

Using one of the gradient recovery methods outlined above, one can provide a recovered Hessian to be used to compute an error estimator for the interpolation error.

In this framework, we present here a mesh adaptation procedure that follows the work of Almeida et al. in [28], extended in [36] for n-dimensional problems.

We consider a triangulation $\mathcal{T}_h$ of the physical domain $\Omega \in \mathbb{R}^d$. If u_h is a finite element approximation of an exact solution u in Ω, then the approximation error can be bound as

$$\|u - u_h\|_{\mathrm{L}^p(\Omega)} \leqslant C\|u - \Pi u\|_{\mathrm{L}^p(\Omega)} \ , \tag{3.14}$$

with a generalization of Equation (3.9), where Π has similar properties as the Clément interpolation operator [37]. It is then possible to use the Hessian of the solution as a bound for this error

$$\|u - \Pi u\|_{\mathrm{L}^p(\Omega)} \leqslant C\|\mathbb{H}(u)\|_{\mathrm{L}^p(\Omega)} \ , \tag{3.15}$$

with

$$\|\mathbb{H}(u)\|_{\mathrm{L}^p(\Omega)} = \left(\sum_{K \in \mathcal{T}_h} \|\mathbb{H}(u)\|^p_{\mathrm{L}^p(A_K)} \right)^{1/p} \ , \tag{3.16}$$

where A_K is a patch formed by K and its neighboring elements.

A recovered Hessian $\mathbb{H}_R$ is provided following the Zienkiewicz–Zhu double gradient recovering technique presented above. Then we have

$$\|\mathbb{H}(u)\|_{\mathrm{L}^p(A_K)} \leqslant C \left(\|\mathbb{H}(u) - \mathbb{H}_R(u_h)\|_{\mathrm{L}^p(A_K)} + \|\mathbb{H}_R(u_h)\|_{\mathrm{L}^p(A_K)} \right) \ . \tag{3.17}$$

We assume that there exists a constant $\overline{C}$ such that

$$\|\mathbb{H}(u) - \mathbb{H}_R(u_h)\|_{\mathrm{L}^p(A_K)} \leqslant \overline{C} N_{\mathcal{T}_h}^{-\beta} \|\mathbb{H}_R(u_h)\|_{\mathrm{L}^p(A_K)} \ , \tag{3.18}$$

with $\beta \geqslant 0$ and $N_{\mathcal{T}_h}$ the number of elements of the triangulation $\mathcal{T}_h$. Combining (3.15) and (3.19) we have

$$\|u - \Pi u\|_{\mathrm{L}^p(\Omega)} \leqslant C \left(1 + N_{\mathcal{T}_h}^{-\beta} \right) \|\mathbb{H}_R(u_h)\|_{\mathrm{L}^p(\Omega)} \ , \tag{3.19}$$

with $\mathbb{H}_R(u_h) = H_R(u_h)(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{x}_K) \cdot (\boldsymbol{x} - \boldsymbol{x}_K)$, where H_R is a recovered Hessian operator and $\boldsymbol{x}_K$ the center of mass of the element K. This shows that the recovered Hessian can be used to control the interpolation error.

If u_h is a good approximation of u we can use the L^p norm of the recovered Hessian as a posteriori error estimator

$$\|u - u_h\|_{\mathrm{L}^p(\Omega)} \approx C' \|H_R(u_h)(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{x}_K) \cdot (\boldsymbol{x} - \boldsymbol{x}_K)\|_{\mathrm{L}^p(\Omega)} \ . \tag{3.20}$$

where C' includes the contribution of C and $(1 + N_{\mathcal{T}_h}^{\beta})$.

To ensure that this approximation will not affect the overall error, and its estimate, we need a reliable stopping criterion to control the convergence of the iterative algorithm. This will be the topic of Section 4.3.

This formulation highlights how the error is not isotropically distributed, and depends on the behavior of the second order derivative of the solution. At this point, it can be of practical interest to introduce a spectral decomposition of H_R, providing a symmetric positive definite matrix

$$\mathcal{H} = R\Lambda R^T \ , \tag{3.21}$$

where R is the orthogonal matrix of the eigenvectors of $\mathbb{H}_R$ and $\Lambda = \text{diag}\{|\lambda_1|, \cdots, |\lambda_d|\}$ is the diagonal matrix of the absolute values of the eigenvalues of $\mathbb{H}_R$. Then we can write the resulting tensor as

$$\mathcal{H} = R\Lambda R^T = |\lambda_1| \boldsymbol{e}_1 \otimes \boldsymbol{e}_1 + \cdots + |\lambda_d| \boldsymbol{e}_d \otimes \boldsymbol{e}_d \ , \tag{3.22}$$

where $\boldsymbol{e}_i$ are the eigenvectors of $\mathbb{H}_R$. If we consider an element $K \in \mathcal{T}_h$, using Relation (3.21) we can derive from Approximation (3.20) the following local error estimator

$$\eta_K^p = \int_K \big(\mathcal{H}(u_h(\boldsymbol{x}_K))(\boldsymbol{x} - \boldsymbol{x}_K) \cdot (\boldsymbol{x} - \boldsymbol{x}_K)\big)^p \ \mathrm{d}\boldsymbol{x} \ . \tag{3.23}$$

Substituting H defined by Relation (3.22) in Equation 3.23 we have

$$\eta_K^p = \int_K \Big(\sum_{i=1}^{d} |\lambda_i(\boldsymbol{x}_K)| [\boldsymbol{e}_i(\boldsymbol{x}_K) \cdot (\boldsymbol{x} - \boldsymbol{x}_K)]^2 \Big)^p \ \mathrm{d}\boldsymbol{x} \ . \tag{3.24}$$

The goal of this adaptation algorithm detailed here, is to use the error estimator as a cost function of an explicit minimization problem, so in the following we will introduce two main properties that are used to provide a simple bound for the estimator:

- The projection of $\boldsymbol{x} - \boldsymbol{x}_K$ on the $\boldsymbol{e}_i$ direction is bounded by

  $$[\boldsymbol{e}_i \cdot (\boldsymbol{x} - \boldsymbol{x}_K)]^2 = \boldsymbol{x}_i^2 \leq h_i^2 \ , \tag{3.25}$$

  then injecting this bound in Equation 3.23

  $$\eta_K^p \leq \int_K \Big(\sum_{i=1}^{d} |\lambda_i(\boldsymbol{x}_K)| h_i^2 \Big)^p \ \mathrm{d}\boldsymbol{x} \ . \tag{3.26}$$

- If the optimal mesh is the one aligned with the solution u, the error has to be locally equidistributed in the principal directions, i.e.

  $$|\lambda_1| h_1^2 = \cdots = |\lambda_i| h_i^2 = \textit{constant} \ . \tag{3.27}$$

  Using this property we can rewrite the bound as

  $$\eta_K^p \leq |K| \big(d |\lambda_d(\boldsymbol{x}_K)| h_d^2\big)^p \ , \tag{3.28}$$

  where $|K|$ is the volume of the element K.

Finally we can define a local indicator using the proposed upper bound

$$\widetilde{\eta}_K \;=\; d|K|^{\frac{1}{p}}|\lambda_d(\boldsymbol{x}_K)|h_d^2 \;. \tag{3.29}$$

This indicator is used as a functional of the following minimization problem

$$\begin{cases} \text{Find } h_K = \{h_{1K}, \cdots, h_{dK}\}, K \in \mathcal{T}_h \text{ that minimizes the cost function ,} \\ \\ F(h_K) = \displaystyle\sum_{K\in\mathcal{T}_h} \widetilde{\eta}_K^p \;, \\ \\ \text{under the constraint} \quad N_{\mathcal{T}_h'} = \; C_0^{-1} \displaystyle\sum_{K\in\mathcal{T}_h} \int_K \prod_{i=1}^{d} \frac{1}{h_{iK}} \;\mathrm{d}\boldsymbol{x} \;, \end{cases} \tag{3.30}$$

where C_0 is the volume of the reference regular simplex, $\mathcal{T}_h'$ is the new triangulation, and the constraint is on the number of nodes $N_{\mathcal{T}_h'}$.

This optimization problem has a unique solution for any $d \geqslant 2$ and the generalized form can be written as

$$\begin{cases} h_{dK} \;=\; \left[\dfrac{d\beta}{(2p+d)\; C_{1K}} \displaystyle\int_K C_{2K} \;\mathrm{d}\boldsymbol{x} \right]^{\frac{1}{2(p+d)}} \;, \\ \\ h_{iK} \;=\; \left(\displaystyle\prod_{k=i}^{d-1} s_{kK} \right) h_{dK} \;, \quad 1 \leqslant i \leqslant d-1 \;, \end{cases} \tag{3.31}$$

where β, C_{1K}, C_{2K} can be computed explicitly, and $s_{iK} = h_i/h_{i+1} = (|\lambda_{i+1}|/|\lambda_i|)^{1/2}$ are the stretching factors for the element K.

Using these results we can now build the metric field that is used by the remeshing procedure

$$\mathrm{M} = \frac{1}{h_1^2}\boldsymbol{e}_1 \otimes \boldsymbol{e}_1 + \cdots + \frac{1}{h_d^2}\boldsymbol{e}_d \otimes \boldsymbol{e}_d \;. \tag{3.32}$$

3.3.3 Edge based estimator and mesh adaptation

We present here a mesh adaptation technique that relies on an error estimator built using the gradient projections on the edges of the elements. The major advantage of this technique is that it does not require any Hessian reconstruction, while it makes use of the projected Hessian on the edges.

We consider an interpolated solution u_h, and we introduce the following notation for a generic node $\mathbf{x}_i$

$$\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i \quad \text{and} \quad \mathrm{u}^{ij} = \mathrm{u}^j - \mathrm{u}^i \quad \forall j \in \Gamma(i) \;. \tag{3.33}$$

The gradient of u_h is a piecewise constant vector field discontinuous on the elements. The projection on the edges of the element, however, is continuous and depends only on the values of u at the extreme nodes of the edges. If we consider a Taylor expansion of the variable of interest on the node $\mathbf{x}_i$

$$\mathrm{u}^j = \mathrm{u}^i + \boldsymbol{\nabla}_{ij} u_h \cdot \mathbf{x}_{ij} \ . \tag{3.34}$$

then using Relation (3.33)

$$\mathrm{u}^{ij} = \boldsymbol{\nabla}_{ij} u_h \cdot \mathbf{x}_{ij} \ . \tag{3.35}$$

and this is true for any element sharing the edge $\mathbf{x}_{ij}$. Considering P_1 elements, we can deduce from the continuity property that the interpolation error on the edges in of second order, as demonstrated in [15, 38]. Then we can formulate the following bound for the error in the projected gradients of the exact and interpolated solutions, using the projected Hessian of the solution

$$|\boldsymbol{\nabla} u_h \cdot \mathbf{x}_{ij} - \boldsymbol{\nabla} u(\mathbf{x}^i) \cdot \mathbf{x}_{ij}| \leqslant \max_{\mathbf{s} \in [\mathbf{x}_i, \mathbf{x}_j]} |\mathbb{H}(u)(\mathbf{s}) \mathbf{x}_{ij} \cdot \mathbf{x}_{ij}| \ , \tag{3.36}$$

where $\mathbb{H}(u)(\mathbf{s})$ is the Hessian of u evaluated at a point $\mathbf{s}$ on the edge $\mathbf{x}_i, \mathbf{x}_j$. Using this inequality we can express the projected Hessian in terms of projected gradient, which is reconstructed on the edges using the nodal values. Since $u|_{\mathbf{x}_{ij}} \in C^2(\mathbf{x}_{ij})$, then $\boldsymbol{\nabla} u|_{\mathbf{x}_{ij}} \in C^1(\mathbf{x}_{ij})$. If we write the Taylor expansion

$$\boldsymbol{\nabla} u(\mathbf{x}_j) = \boldsymbol{\nabla} u(\mathbf{x}_i) + \mathbb{H}(u)(\mathbf{x}_i) \mathbf{x}_{ij} \ , \tag{3.37}$$

from the projection on $\mathbf{x}_{ij}$, we get

$$(\boldsymbol{\nabla} u(\mathbf{x}_j) - \boldsymbol{\nabla} u(\mathbf{x}_i)) \cdot \mathbf{x}_{ij} = \mathbb{H}(u)(\mathbf{x}_i) \mathbf{x}_{ij} \cdot \mathbf{x}_{ij} \ . \tag{3.38}$$

Then we can simplify the notation defining $\mathbf{g}^i = \boldsymbol{\nabla} u(\mathbf{x}_i)$ the gradient of u on a node i, and $\mathbf{g}^{ij} = \mathbf{g}^j - \mathbf{g}^i$ the delta gradient on the edge $\mathbf{x}_{ij}$. Equation (3.38) then becomes

$$\mathbf{g}^{ij} \cdot \mathbf{x}_{ij} = \mathbb{H}(u)(\mathbf{x}_i) \mathbf{x}_{ij} \cdot \mathbf{x}_{ij} \ . \tag{3.39}$$

and we can use this projection to evaluate the interpolation error along the edge

$$e_{ij} = |u - u_h|_{\mathbf{x}_{ij}} = |\mathbf{g}^{ij} \cdot \mathbf{x}_{ij}| \ . \tag{3.40}$$

This is not a proper computable estimator yet, because it implies the use of the gradient of u, that is unknown, and its continuity at the nodes. On P_1 finite elements the gradient is only accessible element-wise, so a gradient recovery procedure is needed. One could use one of the projection methods mentioned in Section 3.3.1, with their limitations mentioned above. The gradient reconstruction proposed here, is based on least square approximation of the gradients on the edges, using the

length distribution tensor. The goal is to find the reconstructed gradient $\mathbf{G}^i \in \mathbb{R}^d$ on each node $\mathbf{x}_i$ such that

$$\mathbf{G}^i = \underset{\mathbf{G}\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{j\in\Gamma(i)} |\left(\mathbf{G} - \boldsymbol{\nabla} u_h\right)\cdot \mathbf{x}_{ij}|^2 = \underset{\mathbf{G}\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{j\in\Gamma(i)} |\mathbf{G}\cdot\mathbf{x}_{ij} - \mathrm{u}^{ij}|^2 \ . \tag{3.41}$$

The minimization problem can be solved imposing the derivative of the argument equal to zero, then

$$\mathbf{G}^i \sum_{j\in\Gamma(i)} \left(\mathbf{x}_{ij}\otimes\mathbf{x}_{ij}\right) = \sum_{j\in\Gamma(i)} \mathrm{u}^{ij}\mathbf{x}_{ij} \ . \tag{3.42}$$

where $\otimes$ identifies the tensor product, and $\mathbb{X}^i = \sum_{j\in\Gamma(i)} \left(\mathbf{x}_{ij}\otimes\mathbf{x}_{ij}\right)$ is the length distribution tensor. To simplify the notation we define

$$\mathbf{U}^i = \frac{1}{|\Gamma(i)|} \sum_{j\in\Gamma(i)} \mathrm{u}^{ij}\mathbf{x}_{ij} \ , \tag{3.43}$$

then we get the recovered gradient on the node $\mathbf{x}_i$

$$\mathbf{G}^i = (\mathbb{X}^i)^{-1}\mathbf{U}^i \ . \tag{3.44}$$

The estimated error in Relation (3.40) now becomes

$$\eta_{ij} = |\mathbf{G}^{ij}\cdot\mathbf{x}_{ij}| \ . \tag{3.45}$$

For a detailed error analysis of the proposed error estimator we refer to [12].

3.4 Numerical examples

Anisotropic mesh adaptation is the perfect tool to capture the anisotropic features of a function of interest. However, when the considered function presents sharp discontinuities the procedure becomes more challenging as the algorithm tends to create elements with extremely high aspect ratios. To obtain the optimal mesh one needs to proceed in an iterative way in order to drive the mesh/solution couple toward optimality, i.e. until the measured interpolation error shows no additional improvement. In this section we present some validation test for the developed edge-based error estimators and anisotropic mesh adaptation technique on two and three dimensional test cases. We present two analytic function to asses the performances of the mesh adaptation on steep gradients and multiscale functions, as well as the flexibility and robustness of the technique applied to the interface capturing of a 3D complex geometry. Finally we perform a comparison of the two error estimators presented above on a simple benchmark.

3.4.1 Modified Holder Table function

As a first numerical example we propose a new benchmark, to show the capability of the adaptation technique to capture a function with steep layers and discontinuities.

We consider the Holder Table analytical function on the domain $[-5,5]\times[-5,5]$

$$f(x,y) = -\left|\sin(x)\cos(y)\exp\left(\left|1-\frac{\sqrt{x^2+y^2}}{\pi}\right|\right)\right| , \tag{3.46}$$

modified imposing upper and lower limiters

$$u(x,y) = \begin{cases} -0.2 & \text{for} \quad f < -0.2 , \\ f & \text{for} \quad -0.2 \leqslant f \leqslant -0.1 , \\ -0.1 & \text{for} \quad f > -0.1 . \end{cases} \tag{3.47}$$

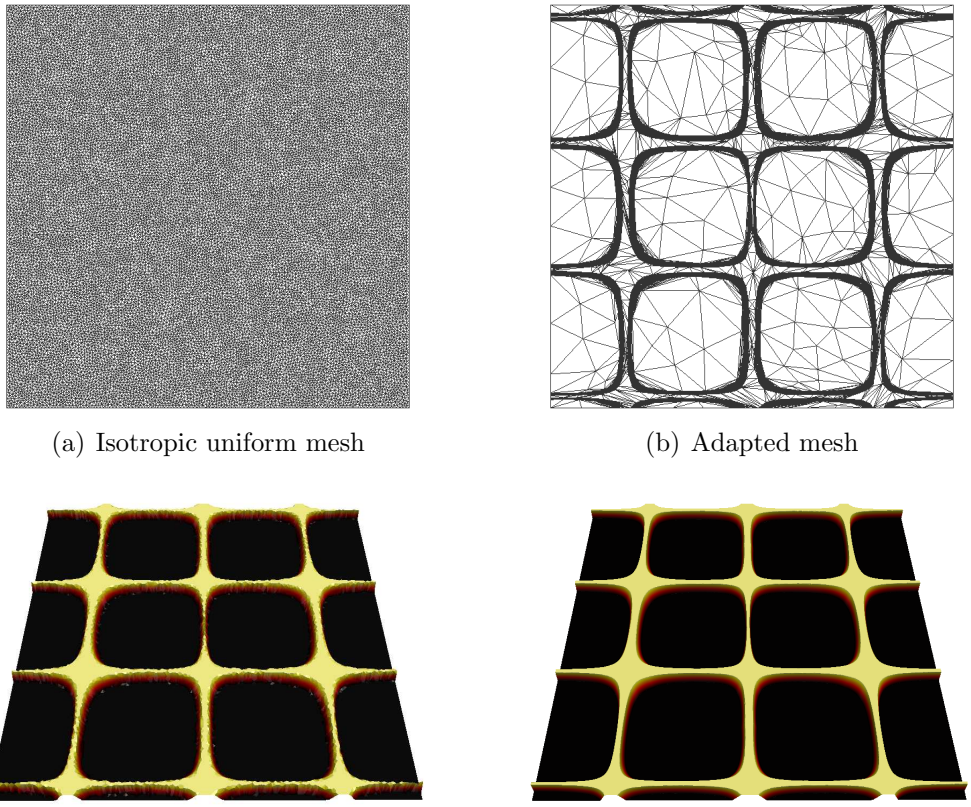
We compare in Figure 3.2 the same analytical function obtained using a 20k nodes on an uniform mesh in Figure 3.2(a) and an anisotropically adapted one in Figure 3.2(b). We can detect how the elements on the adapted mesh are well oriented and stretched along the tangential direction to allow a steep capture of the function, where the uniform mesh fails. The discontinuity are well defined on the adapted mesh and nodes are being automatically redistributed with a higher density in the vicinity of sharp gradients.

3.4.2 Multiscale function

The following test shows the capability of the presented mesh adaptation technique to capture the different scales of the considered function of interest. We consider a multiscale function found in [39] that exhibits variations at small and large scales with respective amplitudes of 0.01 and 1. The analytical function is defined on the domain $[-1,1]\times[-1,1]$ as

$$u(x,y) = \begin{cases} 0.01\sin(50xy) & \text{if} \quad xy \leqslant -\pi/50 , \\ \sin(50xy) & \text{if} \quad -\pi/50 < xy \leqslant 2\pi/50 , \\ 0.01\sin(50xy) & \text{if} \quad 2\pi/50 > xy . \end{cases} \tag{3.48}$$

We can see in Figure 3.3(a) the function evaluated on a 50k nodes anisotropically adapted mesh in Figure 3.3(b). Even if most of the elements are placed on the regions with bigger amplitude, the algorithm can capture the small wiggles as well as the large-scale sinusoidal wave.



(a) Isotropic uniform mesh

(b) Adapted mesh

(c) Function on isotropic uniform mesh

(d) Function on adapted mesh

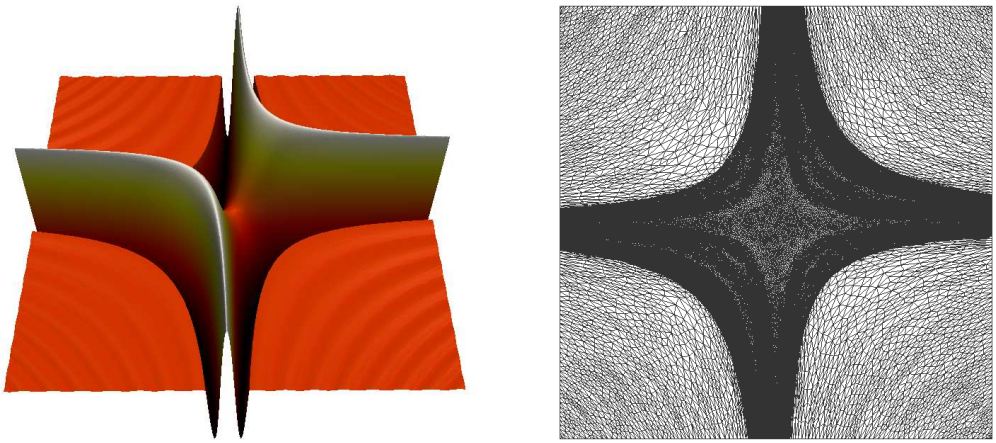
Figure 3.2: Modified Holder Table function, both meshes with 20k nodes

3.4.3 3D solid body

The mesh adaptation algorithm presented above is very flexible and robust to be used for several purposes involving mesh adaptation. When using immersed volume methods [40–43] with a level-set approach [39], it is vital to capture the interface that is implicitly defined as a field function. In Figure 3.4 we can see that the algorithm is capable to capture complex geometries using a number of nodes that is drastically lower compared to a more classic isotropic mesh adaptation. Figure 3.4(c) shows in detail the highly stretched elements on the body interface.

3.4.4 Hessian-Based vs Edge-Based recovery

In this section we evaluate on two manufactured solutions the implementation of the Hessian-based and Edge-based error estimators used for the construction of the discrete metric for topological remeshing: one for the convection–diffusion equation



(a) 3D meshed body (b) Mesh YZ-section

Figure 3.3: 3D mesh adaptation of an immersed solid body

and one for the incompressible Navier–Stokes equations.

3.4.4.1 Convection–Diffusion

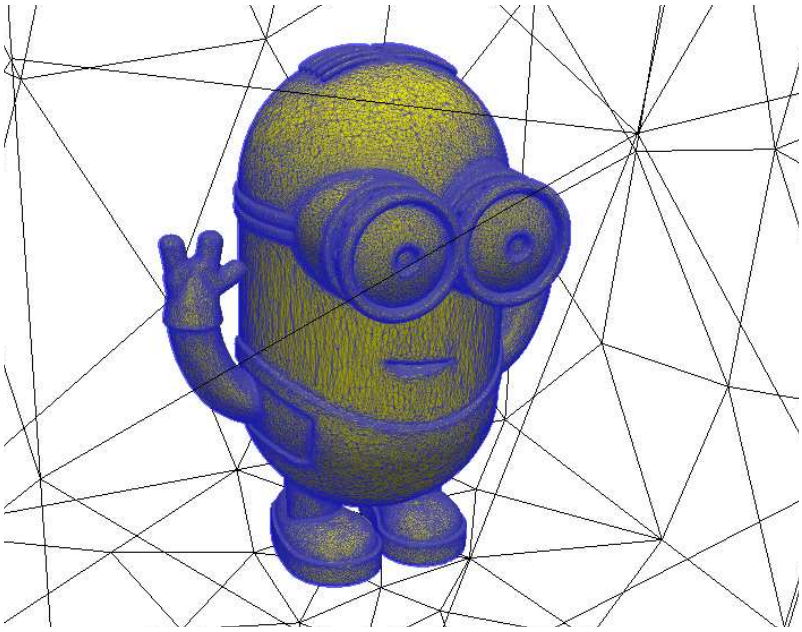
We consider the following problem posed in the unit square $\Omega = [0,1]^2$:

$$\begin{cases} \boldsymbol{v} \cdot \boldsymbol{\nabla} u - \kappa \boldsymbol{\Delta} u = f & \text{in } \Omega \ , \\ u = g & \text{on } \Gamma \ , \end{cases} \tag{3.49}$$

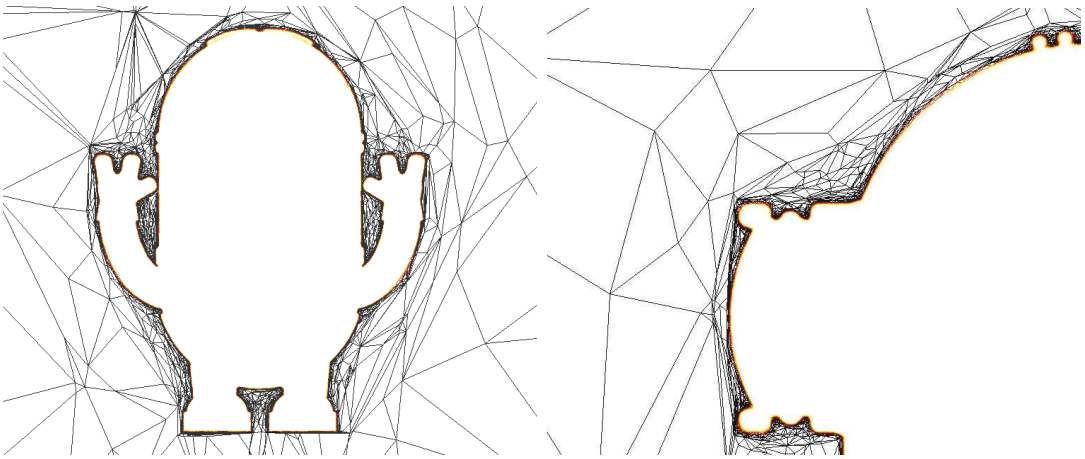
The boundary conditions and the source term are determined so that the exact solution u given by

$$u(x,y) = xy \left(1 - e^{-\frac{1-x}{\kappa}}\right) \left(1 - e^{-\frac{1-y}{\kappa}}\right) \ , \tag{3.50}$$

is solution to the equation, with diffusion coefficient $\kappa = 10^{-3}$ and velocity field $\boldsymbol{v} = (1,1)^T$. The approximate problem is solved on different adapted meshes obtained from the discrete metric based either on a Hessian-based or Edge-based error estimator. The target number of vertices is set successively to 10^2, 10^3, 10^4, then 10^5 vertices. For each mesh the L^2-norm of approximation error is reported in Figure 3.5 and the computational time required for the mesh adaptation procedure is measured and plotted in Figure 3.6.



(a) 3D meshed body



(b) Mesh YZ-section (c) Mesh XZ-section detail

Figure 3.4: 3D mesh adaptation of an immersed solid body

3.4.4.2 Navier–Stokes

We consider an analytic solution of the incompressible Navier–Stokes posed on the unit square:

$$\begin{cases} \rho\left[\partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}\right] + \boldsymbol{\nabla} p - \nu\boldsymbol{\Delta}\boldsymbol{u} = \boldsymbol{f} & \text{in } \Omega\times(0,T)\ , \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 & \text{in } \Omega\times(0,T)\ , \end{cases} \tag{3.51}$$

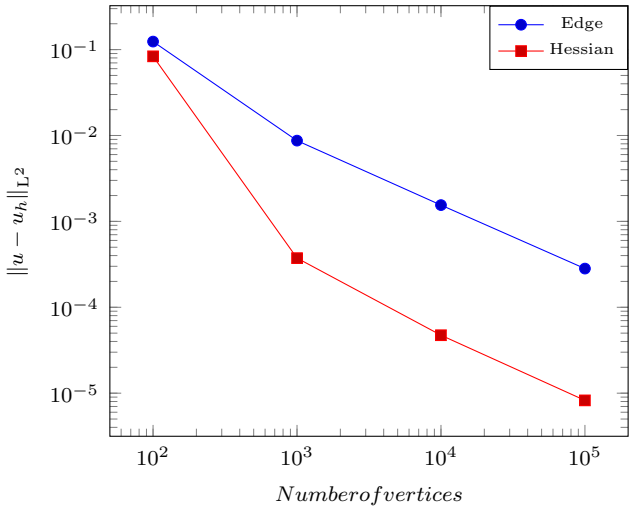


Figure 3.5: L^2*-norm of the error with Hessian-based metric vs Edge-based metric*

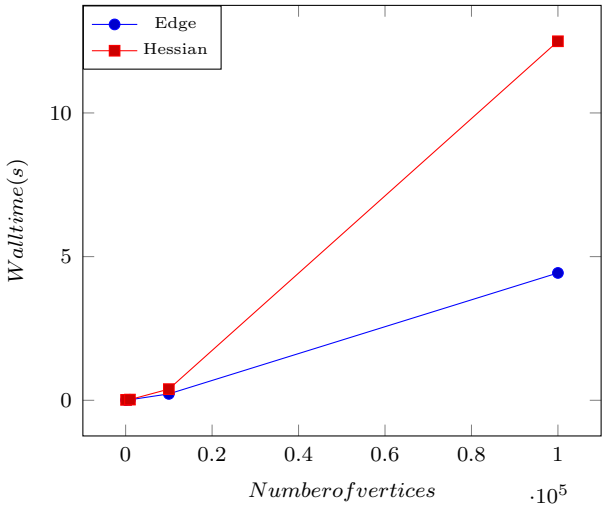


Figure 3.6: Computational time with Hessian-based metric vs Edge-based metric

with the solution pair $(\boldsymbol{u}, p)$ given by

$$\boldsymbol{u}(\boldsymbol{x}, t) = \sin(t)\Big(\sin(\pi x)\sin(\pi y), -\cos(\pi x)\sin(\pi y)\Big)^T$$

and

$$p(\boldsymbol{x}, t) = \sin(\pi x)\sin(\pi y)\cos(t)$$

provided that the corresponding source term is constructed and the Dirichlet boundary conditions are enforced, with parameters $\rho = 1$ and $\nu = 10^-3$. For both the Hessian-based or Edge-based error estimator the target number of vertices is set

successively to 10^2, 10^3, 10^4, then $5 \cdot 10^4$ vertices and the time step was set to 10^4 so that the time discretization error is negligible. For each mesh the L^2-norm of approximation error is reported in Figure 3.7 and the computational time required for the mesh adaptation procedure is measured and plotted in Figure 3.8.

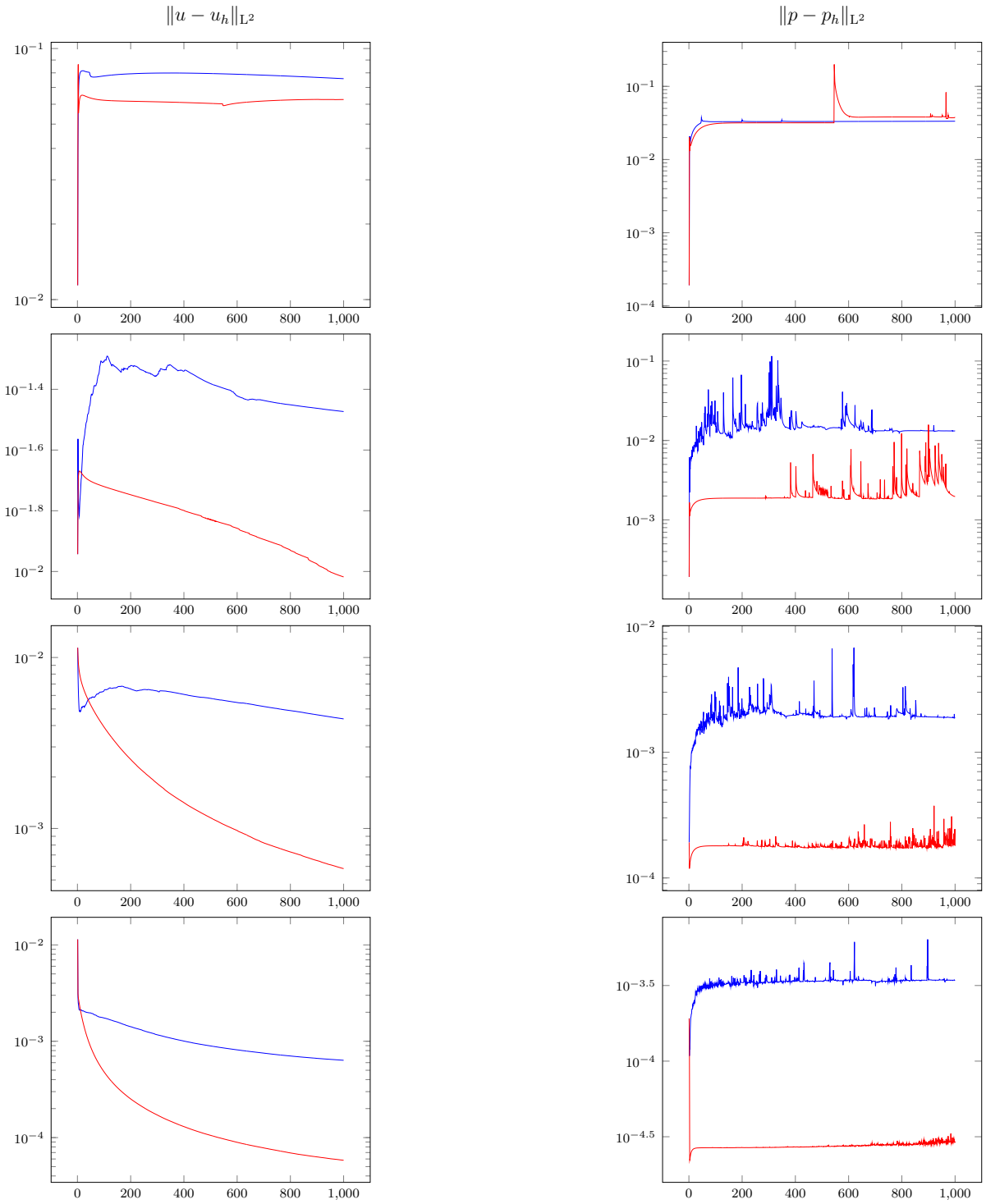


Figure 3.7: Convergence history for the Hessian-based (red) and the Edge-based (blue) estimators for adapted meshes with a target of 10^2, 10^3, 10^4 *and* $5{\cdot}10^4$ *vertices.*

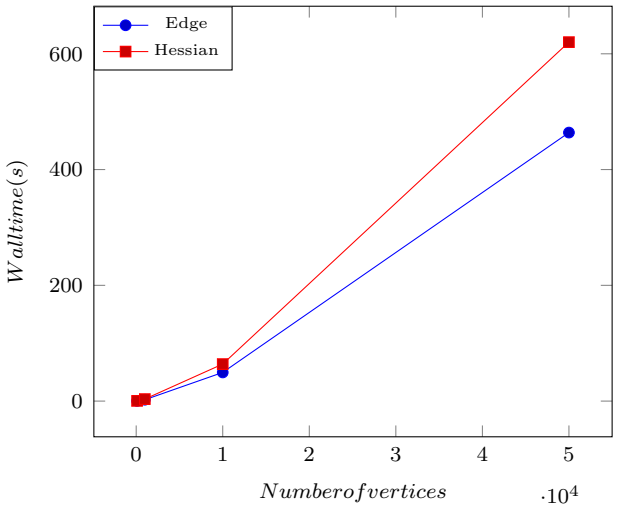


Figure 3.8: Computational time with Hessian-based metric vs Edge-based metric

3.5 Conclusions

In this chapter we briefly introduced the metric-based anisotropic mesh adaptation method used to provide an optimal mesh, that is able to capture steep gradients in the fields of interest. We outline the principles of the metric-based approach, to build a metric tensor in the Riemann space. The result of the procedure is a set of stretching factors that are used to adapt the mesh, resulting in an anisotropic mesh that is well adapted on the field of interest. The adaptation procedure is driven by an error estimator. We propose two a-posterior error estimators, one based on the recovered Hessian of the solution, the second relying on the recovered gradients on the edge of the elements. Both estimators are general and PDE-independent, and can be applied to any field of interest. In the next chapter we propose to use the same estimator, already computed for the mesh adaptation, to drive the convergence of the iterative solution.

3.6 Bibliography

[1] D. C. Arney, J. E. Flaherty, An adaptive mesh-moving and local refinement method for time-dependent partial differential equations, ACM Transactions on Mathematical Software (TOMS) 16 (1) (1990) 48–71. 39

[2] M. J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, Journal of computational Physics 53 (3) (1984) 484–512.

[3] M. Bietermman, J. Flaherty, P. Moore, Adaptive refinement methods for nonlinear parabolic partial differential equations., Tech. rep., Rensselaer Polytechnic Inst. Troy NY Dept. Of Mathematical Sciences (1984).

[4] L. P. Chew, Guaranteed-quality triangular meshes, Tech. rep., Cornell University (1989). 39

[5] J. Peraire, M. Vahdati, K. Morgan, O. C. Zienkiewicz, Adaptive remeshing for compressible flow computations, Journal of computational physics 72 (2) (1987) 449–466. 39

[6] O. C. Zienkiewicz, J. Wu, Automatic directional refinement in adaptive analysis of compressible flows, International Journal for Numerical Methods in Engineering 37 (13) (1994) 2189–2210. 39

[7] D. J. Mavriplis, Adaptive mesh generation for viscous flows using triangulation, Journal of computational Physics 90 (2) (1990) 271–291. 39

[8] P. George, F. Hecht, M. Vallet, Creation of internal points in Voronoi's type method. control adaptation, Advances in engineering software and workstations 13 (5-6) (1991) 303–312.

[9] M. Castro-Díaz, F. Hecht, B. Mohammadi, O. Pironneau, Anisotropic unstructured mesh adaption for flow simulations, International Journal for Numerical Methods in Fluids 25 (4) (1997) 475–491. 39

[10] L. F. Diachin, P. Knupp, T. Munson, S. Shontz, A comparison of two optimization methods for mesh quality improvement, Engineering with Computers 22 (2) (2006) 61–74. 39

[11] J. Kim, T. Panitanarak, S. M. Shontz, A multiobjective mesh optimization framework for mesh quality improvement and mesh untangling, International journal for numerical methods in engineering 94 (1) (2013) 20–42. 39

[12] T. Coupez, A mesh improvement method for 3D automatic remeshing, Numerical Grid Generation in Computational Fluid Dynamics and Related Fields (1994) 615–626. 39, 48

[13] X. Li, M. S. Shephard, M. W. Beall, 3d anisotropic mesh adaptation by mesh modification, Computer Methods in Applied Mechanics and Engineering 194 (48-49) (2005) 4915–4950.

[14] J.-F. Remacle, X. Li, M. S. Shephard, J. E. Flaherty, Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods, International Journal for Numerical Methods in Engineering 62 (7) (2005) 899–923. 39

[15] T. Coupez, Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing, Journal of computational physics 230 (7) (2011) 2391–2405. 39, 47

[16] L. Demkowicz, J. Oden, T. Strouboulis, Adaptive finite elements for flow problems with moving boundaries. i-variational principles and a posteriori estimates, Computer Methods in Applied Mechanics and Engineering 46 (1984) 217–251. 40

[17] R. E. Bank, A. Weiser, Some a posteriori error estimators for elliptic partial differential equations, Mathematics of computation 44 (170) (1985) 283–301. 40

[18] R. E. Bank, R. K. Smith, A posteriori error estimates based on hierarchical bases, SIAM Journal on Numerical Analysis 30 (4) (1993) 921–935. 40

[19] V. Eijkhout, P. Vassilevski, The role of the strengthened Cauchy–Buniakowskii–Schwarz inequality in multilevel methods, SIAM review 33 (3) (1991) 405–419. 40

[20] R. Becker, R. Rannacher, A feed-back approach to error control in finite element methods: Basic analysis and examples, IWR, 1996. 41

[21] J. T. Oden, S. Prudhomme, Goal-oriented error estimation and adaptivity for the finite element method, Computers & mathematics with applications 41 (5-6) (2001) 735–756.

[22] L. Formaggia, S. Perotto, Anisotropic error estimates for elliptic problems, Numerische Mathematik 94 (1) (2003) 67–92. 41

[23] O. C. Zienkiewicz, J. Z. Zhu, A simple error estimator and adaptive procedure for practical engineerng analysis, International journal for numerical methods in engineering 24 (2) (1987) 337–357. 41

[24] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique, International Journal for Numerical Methods in Engineering 33 (7) (1992) 1331–1364. 43

[25] S. Micheletti, S. Perotto, Reliability and efficiency of an anisotropic zienkiewicz–zhu error estimator, Computer Methods in Applied Mechanics and Engineering 195 (9-12) (2006) 799–835. 41

[26] T. Coupez, Génération de maillage et adaptation de maillage par optimisation locale, Revue européenne des éléments finis 9 (4) (2000) 403–423. 41

[27] P. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, Computer Methods in Applied Mechanics and Engineering 194 (48) (2005) 5068 – 5082, Unstructured Mesh Generation. 41

[28] R. C. Almeida, R. A. Feijóo, A. C. Galeão, C. Padra, R. S. Silva, Adaptive finite element computational fluid dynamics using an anisotropic error estimator, Computer Methods in Applied Mechanics and Engineering 182 (3) (2000) 379 – 400. 44

[29] A. Agouzal, Y. V. Vassilevski, Minimization of gradient errors of piecewise linear interpolation on simplicial meshes, Computer Methods in Applied Mechanics and Engineering 199 (33) (2010) 2195 – 2203. 41

[30] A. Bazile, E. Hachem, J. Larroya-Huguet, Y. Mesri, Variational Multiscale error estimator for anisotropic adaptive fluid mechanic simulations: Application to

convection-diffusion problems, Computer Methods in Applied Mechanics and Engineering 331 (2018) 94 – 115. 41

[31] L. Formaggia, S. Micheletti, S. Perotto, Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the Stokes problems, Applied Numerical Mathematics 51 (4) (2004) 511 – 533, applied Scientific Computing: Advances in Grid Generatuion, Approximation and Numerical Modeling. 41

[32] M. Picasso, Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives, Computer Methods in Applied Mechanics and Engineering 196 (1) (2006) 14 – 23. 41

[33] J. Céa, Approximation variationnelle des problèmes aux limites, in: Annales de l'Institut Fourier, Vol. 14, 1964, pp. 345–444. 42

[34] P. Ciarlet, Basic error estimates for elliptic problems, in: Finite Element Methods (Part 1), Vol. 2 of Handbook of Numerical Analysis, Elsevier, 1991, pp. 17 – 351. 42

[35] F. Alauzet, W. Hassan, M. Picasso, Goal oriented, anisotropic, a posteriori error estimates for the laplace equation, in: Numerical Mathematics and Advanced Applications 2009, Springer, 2010, pp. 47–58. 43

[36] Y. Mesri, M. Khalloufi, E. Hachem, On optimal simplicial 3D meshes for minimizing the hessian-based errors, Applied Numerical Mathematics 109 (2016) 235 – 249. 44

[37] P. Clément, Approximation by finite element functions using local regularization, Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique 9 (R2) (1975) 77–84. 44

[38] T. Coupez, G. Jannoun, N. Nassif, H. C. Nguyen, H. Digonnet, E. Hachem, Adaptive time-step with anisotropic meshing for incompressible flows, Journal of Computational Physics 241 (2013) 195–211. 47

[39] A. Loseille, Anisotropic 3D hessian-based multi-scale and adjoint-based mesh adaptation for computational fluid dynamics application to high fidelity sonic boom prediction, Ph.D. thesis, Université Pierre et Marie Curie, Paris VI, Paris, France (2008). 49, 50

[40] D. Lakehal, M. Meier, M. Fulgosi, Interface tracking towards the direct simulation of heat and mass transfer in multiphase flows, International Journal of Heat and Fluid Flow 23 (3) (2002) 242–257. 50

[41] R. Valette, T. Coupez, C. David, B. Vergnes, A direct 3D numerical simulation code for extrusion and mixing processes, International Polymer Processing 24 (2) (2009) 141–147.

[42] J. Bruchon, H. Digonnet, T. Coupez, Using a signed distance function for the simulation of metal forming processes: Formulation of the contact condition and mesh adaptation. From a Lagrangian approach to an Eulerian approach, International journal for numerical methods in engineering 78 (8) (2009) 980–1008.

[43] E. Hachem, T. Kloczko, H. Digonnet, T. Coupez, Stabilized finite element solution to handle complex heat and fluid flows in industrial furnaces using the immersed volume method, International Journal for Numerical Methods in Fluids 68 (1) (2012) 99–121. 50

Chapter 4

An adaptive stopping criterion for iterative solvers

Contents

Résumé *Dans ce chapitre nous proposons un critère d'arrêt adaptatif automatique pour les solveurs itératifs dans le cadre des éléments finis adaptés anisotropes, appliqué aux problèmes dominés par la convection. La formulation proposée tire parti des informations calculées dans la procédure d'adaptation du maillage, qui fournit une estimation de l'erreur d'approximation sans coût de calcul supplémentaire. Nous proposons d'arrêter le solveur itératif lorsque l'erreur algébrique est inférieure au niveau de l'erreur d'approximation estimée. Des tests numériques ont été effectués pour un problème de Laplace symétrique de référence, plusieurs problèmes de convection–diffusion stationnaires et instationnaires, et des problèmes de Navier–Stokes résolus avec des éléments finis stabilisés sur des maillages fortement étirés. Ces cas de test ont été choisis pour leur simplicité de mise en œuvre, pour servir de futures références sur le sujet. Les résultats ont prouvé que, suivant l'approche fournie, le nombre total d'itérations nécessaires peut être réduit jusqu'à cinq fois, sans effet significatif sur la précision de la solution calculée. Il en résulte une réduction considérable du temps de calcul dédié à la résolution itérative du problème. Enfin, un problème modèle de traitement thermique dans un four a été étudié pour démontrer l'applicabilité de l'approche développée à des simulations réalistes à grande échelle. Le solveur numérique couplant Navier–Stokes et l'équation de transfert de chaleur a été exécuté avec des critères d'arrêt adaptatifs basés sur deux estimateurs d'erreur a posteriori. Le temps de calcul s'est avéré favorablement réduit tout en conservant une bonne qualité de la solution discrète par rapport aux quantités d'intérêt : ceci est prometteur pour améliorer les performances des simulations d'applications industrielles en ingénierie.*

4.1 Introduction

When we consider the solution of a system that stems from the finite element discretization of a continuous problem, usually one assumes this solution to be affected only by approximations due to the discretized model. In real-world applications however, this kind of system can be solved efficiently only with an iterative procedure, which introduces another approximation. The accuracy of this approximation is controlled by the stopping criteria used to drive the convergence of the iterative procedure.

Only a small percentage of the research devoted to AFEM aimed to an objective stopping criterion, leaving the user to the dilemma pointed out by Becker et al. in their seminal work [1]:

> *"[...] With no objective stopping criterion available, one has either to continue the iterations until the discrete solution error is practically zero, [...] or take the risk of stopping the iterations prematurely [...]"*

In most cases ad hoc stopping criteria are used, e.g. requiring an initial residual norm to be reduced by a certain factor. These criteria are straightforward to implement, but have no direct link to the actual error in the approximate solution. This could possibly affect the efficiency of the iterative procedure and the accuracy of the resulting solution. On one side an highly accurate approximation is inefficient and most likely unnecessary, on the other side a poor approximation affects the accuracy of the solution and the convergence of the adaptation procedure. This point of view has been developed in several works regarding inexact iterative solvers and stopping criteria. In the framework of symmetric problems, Arioli in [2] proposes an a-priori stopping criterion and Picasso in [3] suggests an a-posteriori approach. Other more general methods, based on a-posteriori flux reconstruction techniques are presented in [4, 5]. In [6], we can find a convergence analysis of different inexact adaptive methods. An interesting overview of the existing approaches is proposed by Arioli and co-workers in [7].

In this work we propose an adaptive stopping criterion, first presented in [8], that follows the strategy developed for symmetric problems in [3], extending the application to non-symmetric problems. We use the general anisotropic mesh adaptation framework presented in Chapter 3, that is shown to be robust and problem independent. Using information from the adaptation procedure we provide a cost-free automatic adaptive control for the linear solver, that proves to be effective to drastically reduce the number of iteration needed, without spoiling at all the accuracy of the solution.

In Section 4.2 we give a brief introduction on iterative solvers and preconditioning techniques, in Section 4.3 we provide the adaptive stopping criterion to control the iterative solver, and finally in Section 4.4 and Section 4.5 we validate this framework with several test cases.

4.2 Iterative solvers for sparse linear systems

4.2.1 Preconditioning

When solving a discrete system in the form

$$\mathrm{A}\mathbf{x} = \mathbf{b} \ , \tag{4.1}$$

with an iterative solver, one parameter to keep under control is the condition number of the matrix of the problem, which is the ratio between the extreme eigenvalues $(\lambda_{\min}, \lambda_{\max})$ of A and denoted by

$$\mathcal{C}(\mathrm{A}) = \frac{\lambda_{\max}}{\lambda_{\min}} \ . \tag{4.2}$$

A suitable method to reduce the condition number of the matrix is needed to have an efficient convergence of the iterative solver. This procedure is called preconditioning and consists in applying a transformation to the initial System (4.1), using a preconditioning matrix.

The preconditioner P is a nonsingular matrix that approximates A, so that the linear system

$$\mathrm{P}^{-1}\mathrm{A}\mathbf{x} = \mathrm{P}^{-1}\mathbf{b} \tag{4.3}$$

has the same solution as System (4.1) but the coefficient matrix has better spectral properties. In this case the system is preconditioned from the left, but one can also precondition from the right

$$\mathrm{A}\mathrm{P}^{-1}\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = \mathrm{P}^{-1}\mathbf{y} \tag{4.4}$$

When Krylov subspace methods are used, computing the preconditioned matrices explicitly would be too expensive and cause a loss of sparsity. To avoid this overhead, matrix–vector products with A and solutions of linear systems of the form $\mathrm{P}\mathbf{z} = \mathbf{r}$ are performed. Another possible choice is split preconditioning

$$\mathrm{P}_1^{-1}\mathrm{A}\mathrm{P}_2^{-1}\mathbf{y} = \mathrm{P}_1^{-1}\mathbf{b}, \quad \mathbf{x} = \mathrm{P}_2^{-1}\mathbf{y} \tag{4.5}$$

where the preconditioner is now $\mathrm{P} = \mathrm{P}_1\mathrm{P}_2$.

Which type of preconditioning to use depends on the choice of the iterative method, problem characteristics, and so forth. If we use GMRES, for example, right preconditioning is often used. In exact arithmetic, the residuals for the right-preconditioned system are identical to the true residuals $\mathbf{r}_k = \mathbf{b} - \mathrm{A}\mathbf{x}_k$ at iteration k of the solver, with $\mathbf{x}_k$ the computed solution vector.

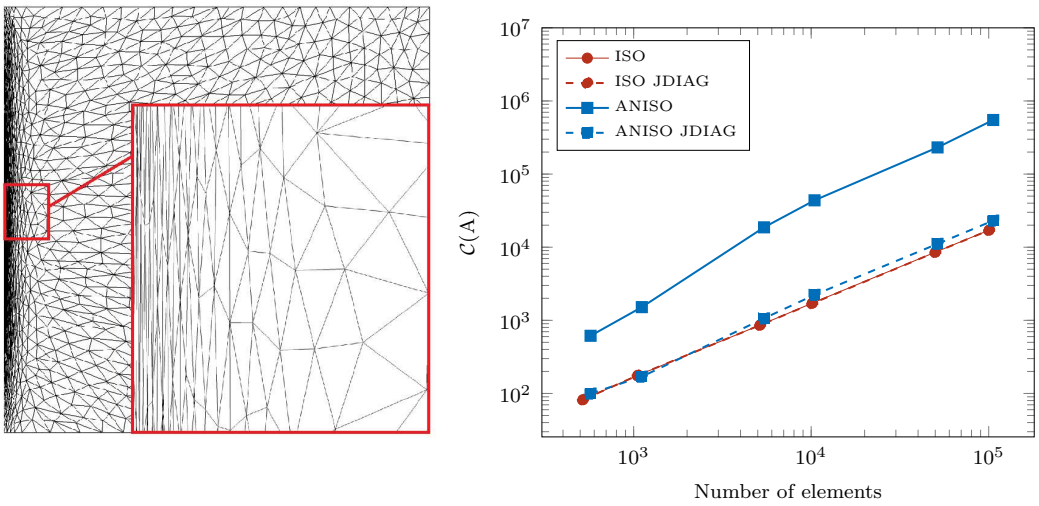
Notice that the matrices $\mathrm{P}^{-1}\mathrm{A}, \mathrm{A}\mathrm{P}^{-1}$, and $\mathrm{P}_1^{-1}\mathrm{A}\mathrm{P}_2^{-1}$ are similar and therefore have the same eigenvalues. However, while for SPD problems the convergence of the Conjugate Gradient (CG) method will be the same in all cases, for nonnormal problems, Krylov solvers like the General Minimal Residual (GMRES) method can behave very differently depending on whether a given preconditioner is applied on the left or on the right. We discuss some of these effects with a numerical example in 4.7.

If we consider anisotropic meshes, we need to take into account also effects of anisotropy on the condition number, that increases on very stretched meshes. Kamenski et al. [9, 10], and Bank and Scott [11] before, show how a simple Jacobi diagonal scaling can be used effectively as a preconditioner for elliptic problems. It is shown that the use of this preconditioner eliminates the effects of mesh non-uniformity in the Euclidean metric, and reduces those caused by the anisotropic adaptation. This indicates that, especially if the mesh concentration is near the boundary, the impact of anisotropy on the performances of the solver is better than

what is commonly feared. Moreover, the impact of mesh anisotropy decreases if the mesh is aligned with the anisotropy of the problem solved. The following tests aim to compare the behavior of anisotropic adapted meshes in terms of conditioning, compared to uniform isotropic meshes, with and without the use of a Jacobi preconditioner.

4.2.1.1 Laplace

First we show this behavior on a Laplace problem on a square domain, with continuous solution and regular boundary layers, using the mesh adaptation algorithm introduced above, with several levels of refinement. We compare the condition number $\mathcal{C}(\mathrm{A})$ of a solution obtained without preconditioning, with the condition number $\mathcal{C}(\mathrm{S}^{-1}\mathrm{AS}^{-1})$ for the solution obtained using a Jacobi diagonal scaling as a preconditioner. An example of the adapted anisotropic mesh obtained with 2000 elements



(a) Anisotropic adapted mesh (2000 elements)

(b) Condition number vs Number of elements

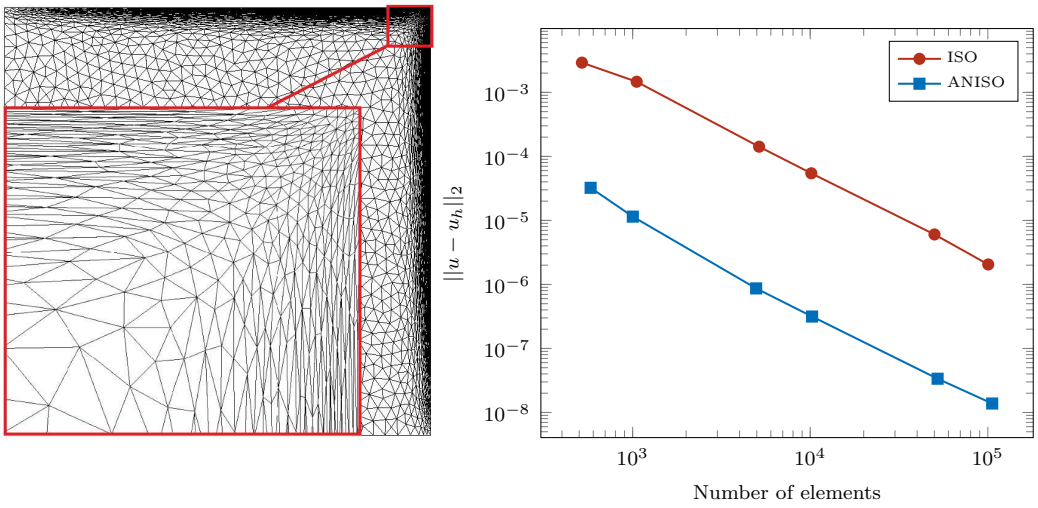
Figure 4.1: Mesh and conditioning for the Laplace problem: dashed lines using diagonal scaling.

is shown in Figure 4.1(a).

In Figure 4.1(b) we show the condition number $\mathcal{C}(\mathrm{A})$ with respect to the number of elements used. While without any preconditioner it is clear that the anisotropic mesh (blue) is ill-conditioned compared to the isotropic one (red), on the other hand, when the Jacobi preconditioner is applied, the conditioning of the two meshes becomes comparable (dashed lines). We can highlight that the diagonalization has no effect of the uniform meshes.

4.2.1.2 Convection–Diffusion

The second test case is a convection–diffusion problem with continuous solution and regular boundary layers. An example of the adapted anisotropic mesh obtained with 5000 elements is shown in Figure 4.2(a). The convergence curves in Figure 4.2(b) show that, using the anisotropic mesh adaptation technique proposed, we can reach higher accuracy with less elements.



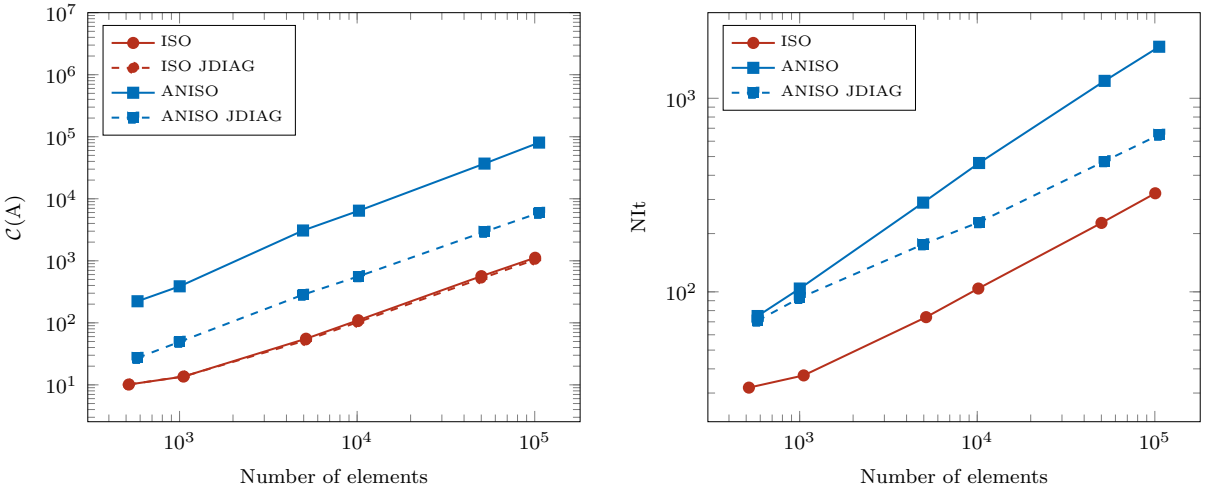
(a) Anisotropic adapted mesh (5000 elements)

(b) L^2 norm of the error vs Number of elements

Figure 4.2: Mesh and convergence curves for the Convection–Diffusion problem.

In Figure 4.3(a) we show the condition number $\mathcal{C}(A)$ with respect to the number of elements used. Without any preconditioner the anisotropic mesh (blue) is very ill-conditioned compared to the isotropic one (red). However, when the Jacobi preconditioner is applied, the conditioning of the anisotropic mesh is highly improved (dashed lines). Figure 4.3(b) shows the impact of conditioning on the number of iterations needed by the linear solver. In this case we consider a GMRES solver with a precision of 10^{-10}. As we can see, the number of iterations needed is reduced by the use of the diagonalization. Moreover, for the anisotropic mesh, this number approaches the value relative to the uniform mesh when we increase the number of elements.

In most of the large scale industrial problems, however, we need to apply a more effective preconditioner to further reduce the conditioning to improve the performance of the iterative solution. ILU (Incomplete LU factorization) preconditioners used with Krylov subspace solvers are the most commonly used methods for solving large-scale spare linear systems [12, 13]. In Figure 4.4 we report the effects of ILU(1) (fill-in factor equal to 1) preconditioning on an isotropic and an anisotropic



(a) Condition number vs Number of elements

(b) Number of iterations vs Number of elements

Figure 4.3: Results for the Convection–Diffusion problem: diagonal scaling in dashed lines

mesh of the convection–diffusion problem. We can see that the conditioning of the

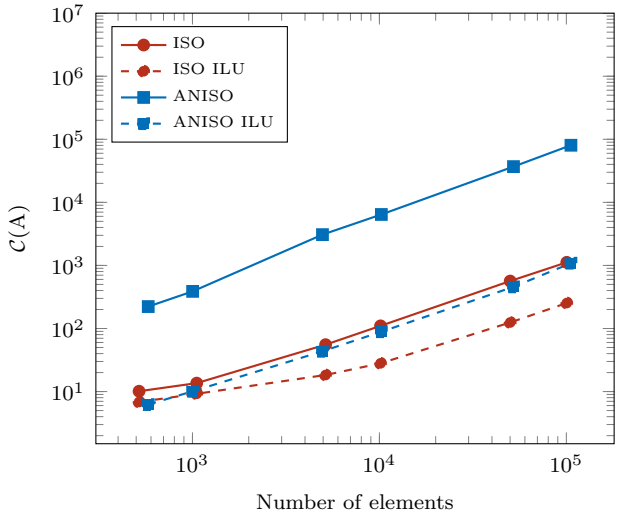
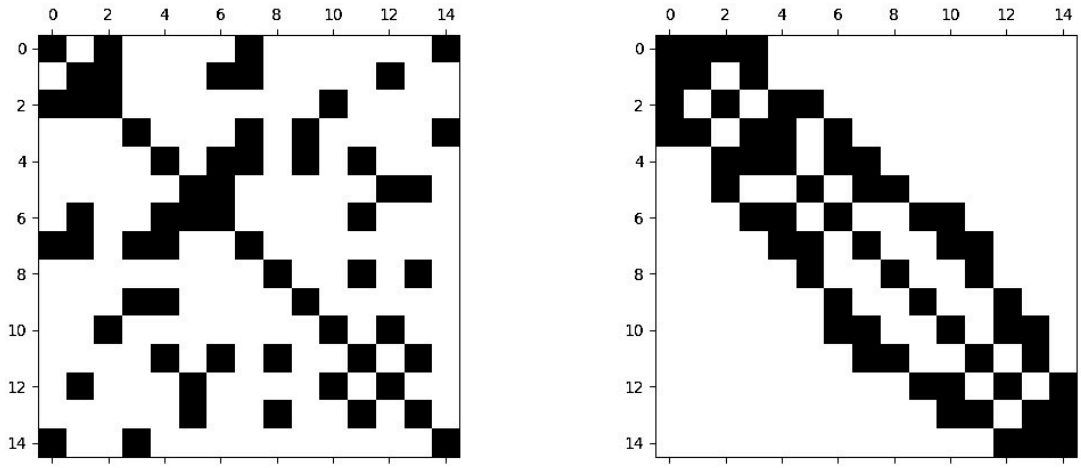


Figure 4.4: Condition number vs Number of elements for the Convection–Diffusion problem

anisotropic mesh matrix preconditioned with ILU (blue dashed) is reduced to the same level of the isotropic mesh (red). In this case if we apply the same preconditioner to the isotropic case (red dashed) we can see that we have some decreasing effect as well.

4.2.2 Matrix reordering techniques

Ordering techniques are frequently used when a matrix is generated with rows and columns numbered according to the natural numbering of the nodes in a mesh. By an appropriate renumbering of the nodes, it is often possible to produce a matrix with a much smaller bandwidth. This can positively affect the rate of convergence of Krylov subspaces iterative solvers, specially with ILU preconditioning. Classical ordering strategies include bandwidth and profile reducing orderings, e.g. the reverse Cuthill-–McKee (RCM) [14], minimum degree orderings [15], and nested dissection [16]. These techniques are based only on the structure of the matrix and not on the numerical values of the matrix entries. The effect of matrix reordering increases significantly on nonsymmetric problems, as pointed out by Dutto in his work on GMRES with ILU(0) preconditioning in the context of solving the compressible Navier–Stokes equations on unstructured grids [17]. In several studies [18, 19] it was found experimentally that RCM gave the best results overall, in terms of performance and robustness, specially on convection-dominated problems. In Figure 4.5 we compare the matrix pattern associated to a simple structured grid, ordering the entries with the natural numbering of the nodes as shown in Figure 4.5(a), and using the RCM reordering technique in Figure 4.5(b). We can see that the reordered



(a) Natural ordering (b) RCM ordering

Figure 4.5: Matrix patterns of a model stiffness matrix

matrix has a much smaller bandwidth, that would lead to a more efficient iterative solution of the system.

Other possible numbering techniques are related to the nature of the problem, such as the downwind numbering for convection-dominated problems proposed in [20].

4.3 Stopping criteria, introduction and existing approaches

When we propose a stopping criterion, the goal is to find a solution u_h^n, obtained at the n_{th} step of the iterative algorithm, that will be a good approximation of the exact solution.

$$\|u - u_h\| \approx \|u - u_h^n\| \ , \tag{4.6}$$

where u_h is the approximated solution ideally obtained with a direct solver, and with $\|\cdot\|$ we identify a certain norm (or seminorm).

The main ingredients needed to build a stopping criterion are: (i) the estimate of the algebraic error $\|u_h - u_h^n\|$, (ii) the formulation of the quantity used to bound it, e.g., an estimation of the approximation or interpolation error, (iii) a suitable scale factor for the aforementioned bound.

In the following we outline several possible approaches found in the literature. One main difference is the choice made to estimate the error used as a bound for the algebraic error: a-priori estimators are based only on the information available before the solution of the system, while a-posteriori estimators take advantage of the computed solution.

4.3.1 A-priori stopping criterion for elliptic problems

Symmetric and positive definite systems, resulting from the finite element formulation of elliptic PDE, are commonly solved using the Conjugate Gradient (CG) algorithm. Using CG, it is quite natural to have a stopping criterion which takes advantage of the minimization property of on the energy norm of the error, that comes with this method. Arioli, in [2], proposed a stopping criterion based on this evaluation of the energy norm of the error, following an *a priori* approach. We define the energy norm $\|\cdot\|_A$ as the A-conjugate scalar product, where A is the SPD matrix of the linear system corresponding to the symmetric coercive bilinear form $a(\cdot,\cdot)$ of the problem considered,

$$\|\mathbf{y}\|_{\mathrm{A}} = (\mathbf{y}^T \mathrm{A}\mathbf{y})^{1/2} \ , \tag{4.7}$$

with $\|\cdot\|_{\mathrm{A}^{-1}}$ we identify the dual norm

$$\|\mathbf{f}\|_{\mathrm{A}^{-1}} = (\mathbf{f}^T \mathrm{A}^{-1}\mathbf{f})^{1/2} \ , \tag{4.8}$$

If we define $\mathbf{r}_n = \mathbf{b} - \mathrm{A}\mathbf{u}_n$ the residual at step n, we can write

$$\|\delta\mathbf{u}_n\|_{\mathrm{A}} = \|\mathbf{r}_n\|_{\mathrm{A}^{-1}} \ . \tag{4.9}$$

Therefore a stopping criterion can be defined as

$$\|\mathrm{A}\mathbf{u}_n - \mathbf{b}\|_{\mathrm{A}^{-1}} \leq \varepsilon \|\mathbf{b}\|_{\mathrm{A}^{-1}} \ . \tag{4.10}$$

If the criterion is satisfied, the solution vector $\mathbf{u}_n$ is the exact solution of the perturbed linear system $\mathrm{A}\mathbf{u}_n = \mathbf{b} - \mathbf{r}_n$, so that

$$\|\mathbf{u} - \mathbf{u}_n\|_{\mathrm{A}} = \|\mathbf{r}_n\|_{\mathrm{A}^{-1}} \ , \tag{4.11}$$

then

$$\|\mathbf{r}_n\|_{\mathrm{A}^{-1}} \leq \varepsilon \|\mathbf{b}\|_{\mathrm{A}^{-1}} = \varepsilon \|\mathbf{u}\|_{\mathrm{A}} \ . \tag{4.12}$$

The scaling factor $\varepsilon < 1$ is user defined: a practical choice could be $\varepsilon \approx h^2$ or $\varepsilon \approx h$.

To provide a computable implementation of this criterion, an estimation of the residual norm at the LHS is needed. This can be provided using several methods, e.g. the Gauss quadrature rules proposed in [21] or using the formulation suggested by Hestenes and Stiefel (HS) in [22]. The latter is straightforward to compute using quantities already computed in the CG iteration. The assumption is that the algebraic error decreases along the CG steps, so $e_{alg}^{n+d} \ll e_{alg}^{n}$, with d a suitable delay. The estimate of the error at the step n will be:

$$\eta_{alg}^{n} = \sum_{j=n+1}^{n+d} \alpha_j \|\mathbf{r}_j\|^2 \ , \tag{4.13}$$

the value of d needed for a reliable estimation depends on the regularity of the solution u, and $\{\alpha_j\}$ denote the quadrature weights.

The energy norm of the solution can be bounded taking advantage of the minimization properties of the CG method. The energy norm of the error $\delta\mathbf{u}_n$ is minimized at each step n on a Krylov space. Due to this property we can write that

$$\|\mathbf{u}\|_{\mathrm{A}}^2 \geq \mathbf{u}_n^T \mathbf{r}_0 + \mathbf{b}^T \mathbf{u}_0 \ . \tag{4.14}$$

Replacing this bound in Relation (4.10) we obtain the final formulation of the stopping criterion

$$\xi \leq \varepsilon^2 \left(\mathbf{u}_n^T \mathbf{r}_0 + \mathbf{b}^T \mathbf{u}_0 \right) \ . \tag{4.15}$$

The main drawback of the method is the need to compute additional unneeded iterations, and above all the choice of the d parameter that is not trivial for complex problems. Figure 4.6 shows that on a L-shaped domain problem, only for $d \geq 90$ the oscillations of the estimates are small enough to provide an accurate solution.

4.3.2 A-posteriori stopping criterion for elliptic problems

In [3] Picasso suggests a practical implementation of an adaptive stopping criterion for the CG algorithm, used to solve elliptic problems. The method proposed, compares the energy norm of the algebraic error with the error estimator used for

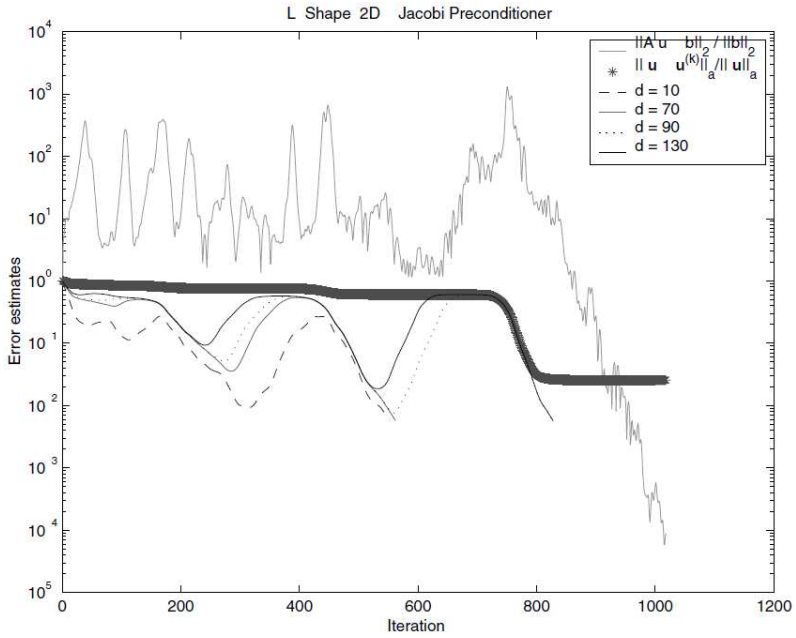


Figure 4.6: Comparison of several estimates of the energy error for an L-shaped domain problem [2].

the adaptation. The iterative procedure is stopped when the first is small enough, compared to the second.

First the author provides a new version of the anisotropic estimator, introduced in [23], for the energy norm of the error $\|u - u_h^n\|$

$$\eta^2 = \sum_{K \in \mathcal{T}_h} \eta_K^2 \qquad \text{with} \qquad \eta_K^2 = \rho_K \omega_K(e) \ , \tag{4.16}$$

where $\omega_K(e)$ is an Hessian based anisotropic factor and ρ_K is a residual based isotropic scaling. The anisotropic contribution relies on the approximation for the error derivatives suggested by Zhu–Zienkiewicz in [24]. The two following inequalities provide a bound for the error $u - u_h^n$, using the estimator η, including the algebraic error contribution, that is used to control the convergence of the iterative solution.

$$\begin{aligned} \int_\Omega |\boldsymbol{\nabla} e|^2 \, \mathrm{d}\boldsymbol{x} \leq & C_1 \sum_{K \in \mathcal{T}_h} \eta_K^2 + \int_\Omega \left| \boldsymbol{\nabla} \left(u_h - u_h^n \right) \right|^2 \mathrm{d}\boldsymbol{x} + h.o.t. \ , \\ & \sum_{K \in \mathcal{T}_h} \eta_K^2 \leq C_2 \int_\Omega |\boldsymbol{\nabla} e|^2 \, \mathrm{d}\boldsymbol{x} + h.o.t. \ . \end{aligned} \tag{4.17}$$

The two following inequalities allow to relate the error estimator η to the true error $u - u_h^n$ and to the error allowing to stop the CG algorithm after n iterations,

$u_h - u_h^n$ (error related to the algebraic resolution.) where $h.o.t$ is a high-order term bounded by the aspect ratio of meshes and data. The error estimator η can be, then, considered to be equivalent to the true error $u - u_h^n$.

Using these ingredients, the CG algorithm is stopped when the energy norm of the algebraic error is lower than a fraction of the estimated error

$$\|\mathbf{u} - \mathbf{u}_n\|_A \leq 0.01 \left(\sum_{K \in \mathcal{T}_h} \eta_K^2 \right)^{1/2} , \tag{4.18}$$

where the choice of the scaling factor at 1% is purely heuristic.

Using the definition of the energy norm, we can write $\|\mathbf{u}-\mathbf{u}_n\|_A = \|\boldsymbol{\nabla} (u_h - u_h^n)\|_{\mathrm{L}^2}$ as

$$\|\boldsymbol{\nabla} (u_h - u_h^n)\|_{\mathrm{L}^2} = (\mathbf{u} - \mathbf{u}_n)^T A (\mathbf{u} - \mathbf{u}_n) = \mathbf{r}_n^T A^{-1} \mathbf{r}_n \ . \tag{4.19}$$

If A is a symmetric positive definite matrix, then it defines the A-conjugate inner-product. If we consider anisotropic meshes the author suggests that this quantity can be approximated with the Euclidean norm of the residual (further investigation in Section 4.4).

$$\|\mathbf{r}_n\|_{A^{-1}} \approx \|\mathbf{r}_n\| \ . \tag{4.20}$$

Then the stopping criterion for the CG iteration can be defined as

$$\|\mathbf{r}_n\| \leq 0.01 \left(\sum_{K \in \mathcal{T}_h} \eta_K^2 \right)^{1/2} . \tag{4.21}$$

The stopping criterion proposed is well integrated in the framework of the adaptive finite element method, and shows good performances in the reduction of iterations and computational time. However, it would be interesting to extend this approach to parabolic and non-symmetric problems

4.3.3 Stopping criteria based on comparing different error components

A general approach to the subject is provided by Ern and Vohralik [4]. The basic idea is always the same, extended to all the main sources of error in the finite element model. All the error should be lower than the one that dominates.

The study analyses the case of non-linear PDEs, providing an algorithm to control the non-linear and linear iterations. The criteria are based on a-posteriori error estimates that separate the different error sources. For each source – namely the discretization error, the linearization error, and the algebraic error – an estimator is

provided. The method developed is based on the balanced flux reconstruction technique, see Prager and Synge [25]. With all the error sources available, it is possible to control the linear and non-linear iterations to maintain a similar level of accuracy. The algebraic iterations are stopped when the discretization or linearization error dominates. In the same spirit, the non-linear iterations are stopped when the discretization error component starts to dominate. A similar approach is followed by Jiranek et al. in [5].

The error estimator is given as the sum of the different contributions

$$\eta_{tot} = \eta_{disc} + \eta_{lin} + \eta_{alg} + \eta_{rem} \tag{4.22}$$

Where η_{disc} stands for the discretization error, η_{lin} represents the linearization error and η_{alg} the algebraic error. η_{rem} is an error component from the algebraic remainder, a function that stems from the formulation of the quasi-balanced flux reconstruction. Using this formulation, the approximation of the flux is more practical to compute. The presence of the algebraic reminder requires an extra constraint to ensure that this function is small enough.

The proposed algorithm reads as follows:

1: Choose an initial solution $\mathbf{u}^0 \in \mathbb{R}^N$, set $k = 1$ (non-linear iterations counter)
2: **loop** *non-linear*
3: Set up the linear algebraic system $\mathrm{A}_{k-1}\mathbf{u}_{k-1} = \mathbf{f}_{k-1}$
4: Define $\mathbf{u}_k^0 = \mathbf{u}_{k-1}$ and set $n = 0$ (linear iterations counter)
5: **loop** *linear*
6: Perform $\nu > 0$ steps of the linear solver for $\mathbf{u}_k^n$ (increase ν progressively)
7: **if** $\eta_{rem}^{k,n} \leq \gamma_{rem} \max\left[\eta_{disc}^{k,n}, \eta_{lin}^{k,n}, \eta_{alg}^{k,n}\right]$ and $\eta_{alg}^{k,n} \leq \gamma_{alg} \max\left[\eta_{disc}^{k,n}, \eta_{lin}^{k,n}\right]$ **then**
8: set $\mathbf{u}_k = \mathbf{u}_k^n$
9: **end loop** *linear*
10: **else**
11: set $n = n + \nu$
12: **go to** 5
13: **if** $\eta_{lin}^{k,n} \leq \gamma_{lin}\eta_{disc}^{k,n}$ **then**
14: **end loop** *non-linear*
15: **else**
16: set $k = k + 1$
17: **go to** 2

---

where γ_i are scaling parameters usually ~ 0.1.

The additional steps ν performed in the linear solver are needed to build the algebraic reminder and the flux reconstruction of the algebraic residual.

The method proposed is very general and takes into account all the principal sources of error in the model. The tests performed by the authors suggest good performances in the reduction of the computational time needed by the solver. The construction of all the estimator, however, can be computationally demanding, and there are a lot of scaling factors to be tuned. The authors themselves suggest the possibility to compute the estimate only periodically or consider simplifications.

4.4 Adaptive stopping criterion for CDR equations

In the next section we detail the new stopping criterion at the base of this work. The idea is that all the sources of error should be lower then the one that dominates, generally the discretization error. If we provide a suitable estimator of the discretization error $\|u - u_h\|$, we can use it as a stopping criterion compared to an estimator of the algebraic error $\|u_h^n - u_h\|$.

In the light of the adaptive framework presented in Chapter 3, we can see that in the presented mesh adaptation procedure we compute a suitable candidate for the estimator of the discretization error. This implies that we already have the first ingredient for the stopping criterion, without additional computational effort.

The method proposed here is general and independent from the estimator used, that has to be considered as an indicator of the order of magnitude of the biggest source of error in our solution process. This method can be considered as an extension of the one introduced in [3] for elliptic problems. To cope with the nature of convection-dominated CDR equations, we decide here to use L^p-norms for the error. This choice is line with both the estimators proposed in Chapter 4, and is due to the fact that L^p-norms are more general for this kind of problems, unlike the energy norms, that are specific to elliptic problems.

If we consider the local error estimator η_K that approximates the error in the L^2-norm $\|u - u_h\|_{\mathrm{L}^2(K)}$, we can define the global stopping criterion as a bound for the algebraic error on the discretized domain Ω

$$\|u_h - u_h^n\|_{\mathrm{L}^2(\Omega)} \leqslant c \left(\sum_{K \in \mathcal{T}_h} \eta_K^2 \right)^{1/2} , \tag{4.23}$$

where c is a user defined scaling factor that defines the error reduction we want to impose.

However, to have a computable stopping criterion we need some estimate of the algebraic error norm. In general a reliable estimate of this quantity can be non trivial to provide. Arioli in [26] presents a review of several techniques that can

be used for this purpose, in the framework of symmetric problems. In the same framework, as mentioned in Section 4.3.2, Picasso proposes the euclidean norm of the residual $\|\mathbf{r}_n\|$ as a good approximation for the norm of algebraic error. Where the residual vector $\mathbf{r}_n$, following from the linear system, is defined as

$$\mathbf{r}_n = \mathbf{f} - \mathrm{A}\mathbf{u}_n \ . \tag{4.24}$$

Supporting this thesis we performed some tests on the comparison between the algebraic error norm and the residual norm for the Laplace problem. As shown in Figure 4.7, we found the same results presented in **??** for the energy norm of the algebraic error. We can see that, as expected, the use of the energy norm is the natural choice for this kind of problems. In our framework, however, we choose to use the L^2-norm for the aforementioned reasons. This choice translates in an underprediction of the residual norm of roughly one order of magnitude. To compare with the results in the literature on this test case, we adjust consequently the choice of the c constant in the stopping criterion.

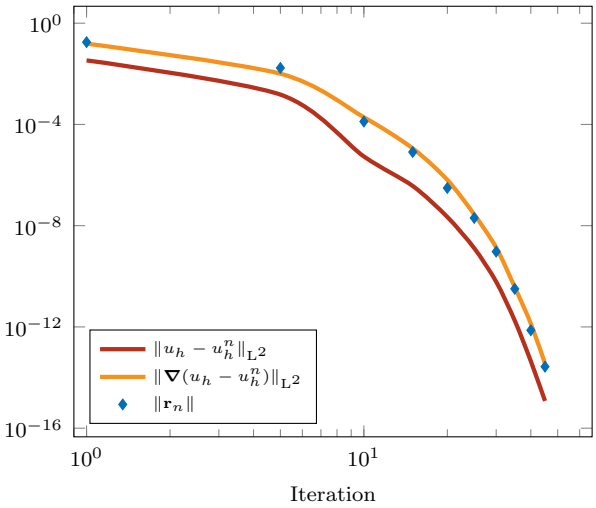


Figure 4.7: Comparison between algebraic error and residual norm for test case 4.4.1.1; solution on the final adapted mesh.

For the Convection–Diffusion problem we performed similar tests to asses the reliability of this approximation. In Figure 4.8 we show the euclidean norm of the residual computed at each GMRES iteration compared to the algebraic error in L^2 norm, that is naturally minimized by the GMRES algorithm. The results suggest that after a few initial iterations the euclidean norm of the residual can be considered as a good approximation for the L^2 norm of the algebraic error.

After these considerations the stopping criterion defined at Inequality (4.23)

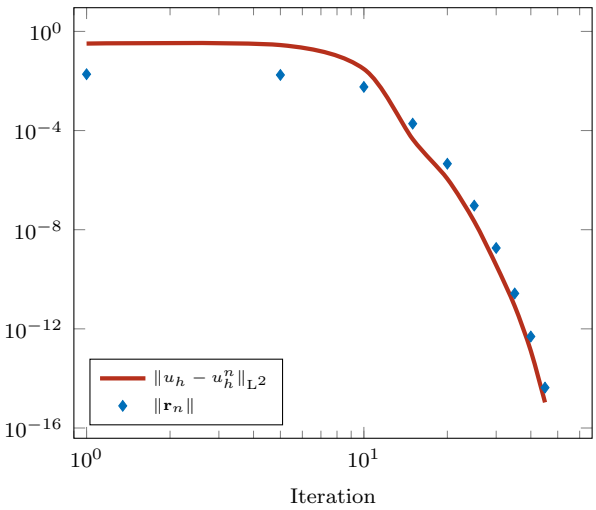


Figure 4.8: Comparison between algebraic error and residual norm for test case 4.4.1.2; solution on the final adapted mesh.

becomes

$$\|u_h - u_h^n\|_{L^2} \approx \|\mathbf{r}_n\| \leqslant c \left(\sum_{K \in \mathcal{T}_h} \eta_K^2 \right)^{1/2} . \tag{4.25}$$

In the following we will use this stopping criterion for the solution of several test cases, comparing the result obtained using a classical approach with a fixed stopping criterion.

4.4.1 Application to model problems

In this section we apply the proposed adaptive stopping criterion to the solution of several test cases. The aim is to validate the method with known problems, and propose new benchmarks for the CDR equation. We compare the new adaptive

$$\|\mathbf{r}_n\| \leqslant c \left(\sum_{K} \eta_K^2 \right)^{1/2} , \tag{4.26}$$

with a classical fixed stopping criterion, where we impose a given precision, e.g., $\epsilon = 10^{-10}\|\mathbf{f}\|$,

$$\|\mathbf{r}_n\| \leq 10^{-10}\|\mathbf{f}\| . \tag{4.27}$$

We first consider several steady test cases. The first benchmark is a simple Laplace problem, then we consider one convection–diffusion problem with analytic solution and one with a domain that generates a singularity. For each adaptive

computation, 30 adaptation steps are performed using the anisotropic adaptive procedure presented in Section 3.2, providing final adapted meshes with extremely high aspect ratios. For each step of adaptation, the interpolated solution computed on the previous mesh is used as an initial guess for the linear solver. The error estimator used for the stopping criterion is also computed using the solution obtained on the previous adapted mesh.

The last test case of this section is a time dependent convection–diffusion problem, with challenging anisotropic features, exhibiting evolving internal and boundary layers. The adaptive simulation is performed with one step of mesh adaptation at each time step, and the error estimator used for the stopping criterion is computed using the solution obtained at the previous step.

4.4.1.1 Laplace Problem

We consider as a first reference the Laplace problem in the unit square $\Omega = [0,1]^2$, proposed in [3, 27]

$$\begin{cases} -\boldsymbol{\Delta} u = f, & \text{in } \Omega \ , \\ u = 0, & \text{in } \Gamma \ . \end{cases} \tag{4.28}$$

We choose f so that the exact solution u is described by the following analytical formulation

$$u(x,y) = 4(1 - e^{-ax} - (1 - e^{-a})x)y(1-y) \ , \tag{4.29}$$

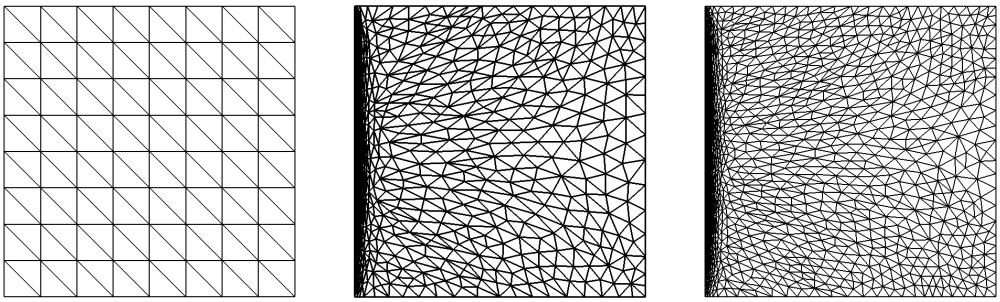
with $a = 10^2$.

The linear system that stems from the finite element discretization of this elliptic problem is characterized by an SPD stiffness matrix. Exploiting this property, a CG algorithm is used to compute the solution, with an Incomplete LU factorization (ILU) as a preconditioner.

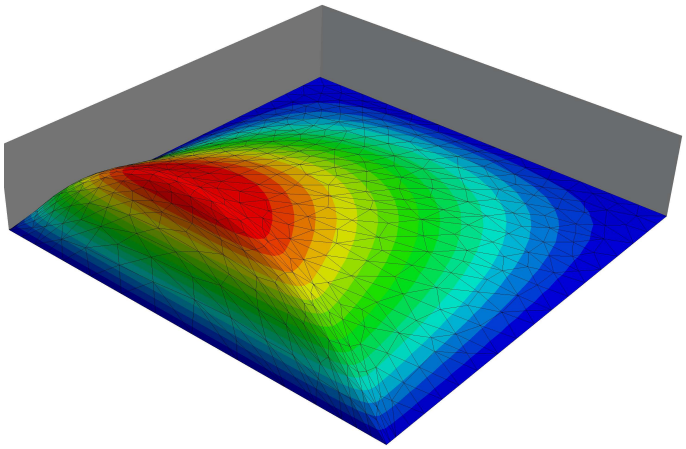
In Figure 4.9 we show several adaptation steps from the initial mesh Figure 4.9(a) to the final mesh Figure 4.9(c), and the contours of the computed solution on the final mesh Figure 4.9(d).

In Table 4.1 we reported the results of the convergence analysis for four different adaptations with increasing refinement, where $N_{\mathcal{T}_h}$ is the number of vertices in the final mesh, and N_{alg} is the total number of linear solver iterations.

As we can see in Figure 4.10(a), when using the proposed adaptive stopping criterion (4.25), the total number of CG iterations considerably decreases. We can also see that the results obtained using our error estimator to build the adaptive stopping criterion are in qualitative agreement with those obtained by Picasso [3] on the same test case but with a different error estimator. In Figure 4.10(b) we show that the use of the proposed stopping criterion does not affect the convergence of the method with respect to the number of nodes used, resulting in the same accuracy between the compared results.



(a) Initial mesh (b) 15th adaptation step (c) 30th adaptation step



(d) Final mesh and computed solution contours, 16 values between 0.0 and 0.9

Figure 4.9: Adaptation steps with final computed solution

4.4.1.2 Convection–Diffusion, analytic solution

After assessing the method with an elliptic problem we move to the more challenging convection–diffusion problem (4.30), that we consider in the unit square $\Omega = [0,1]^2$:

$$\begin{cases} \boldsymbol{v} \cdot \boldsymbol{\nabla} u - \kappa \boldsymbol{\Delta} u = f & \text{in } \Omega \ , \\ u = g & \text{on } \Gamma \ , \end{cases} \tag{4.30}$$

The boundary conditions and the source term are determined so that the exact solution u is described by the following analytical formulation

$$u(x,y) = xy\left(1 - e^{-\frac{1-x}{\kappa}}\right)\left(1 - e^{-\frac{1-y}{\kappa}}\right) \ , \tag{4.31}$$

with diffusion coefficient $\kappa = 10^{-3}$ and velocity field $\boldsymbol{v} = (1,1)^T$, [28, 29]. The linear system that stems from the finite element discretization of this problem is solved

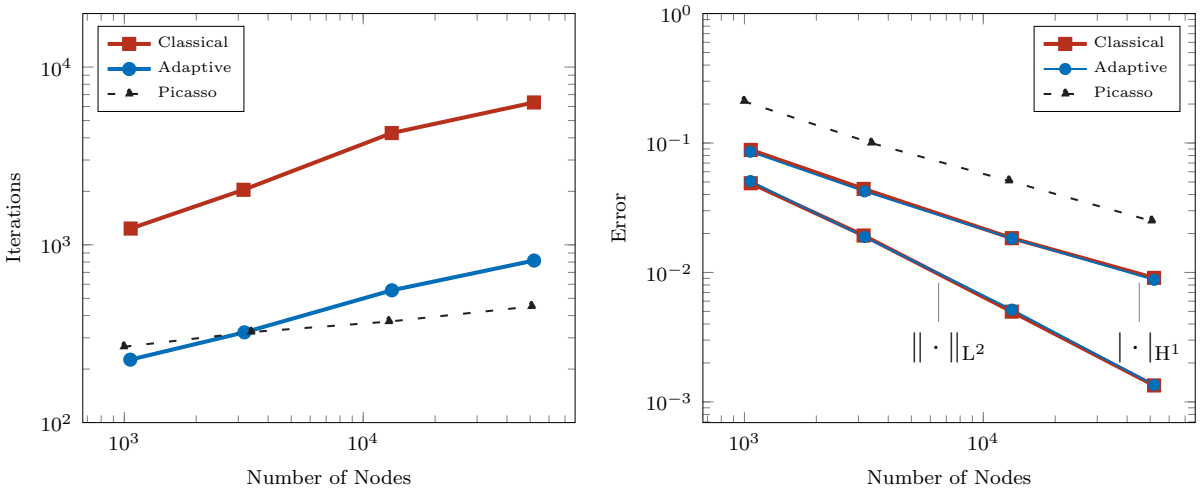
Table 4.1: Results for four different adaptations

(a) Classical stopping criterion

$N_{\mathcal{T}_h}$	N_{alg}	$\|\nabla(u-u_h^n)\|_{\mathrm{L}^2}$	$\|u-u_h^n\|_{\mathrm{L}^2}$
1063	1235	$8.84\cdot10^{-2}$	$4.89\cdot10^{-2}$
3155	2040	$4.42\cdot10^{-2}$	$1.93\cdot10^{-2}$
13166	4254	$1.84\cdot10^{-2}$	$4.98\cdot10^{-3}$
51865	6322	$9.08\cdot10^{-3}$	$1.34\cdot10^{-3}$

(b) Adaptive stopping criterion

$N_{\mathcal{T}_h}$	N_{alg}	$\|\nabla(u-u_h^n)\|_{\mathrm{L}^2}$	$\|u-u_h^n\|_{\mathrm{L}^2}$
1059	226	$8.56\cdot10^{-2}$	$5.08\cdot10^{-2}$
3184	322	$4.23\cdot10^{-2}$	$1.88\cdot10^{-2}$
13188	555	$1.83\cdot10^{-2}$	$5.16\cdot10^{-3}$
51947	816	$8.77\cdot10^{-3}$	$1.35\cdot10^{-3}$



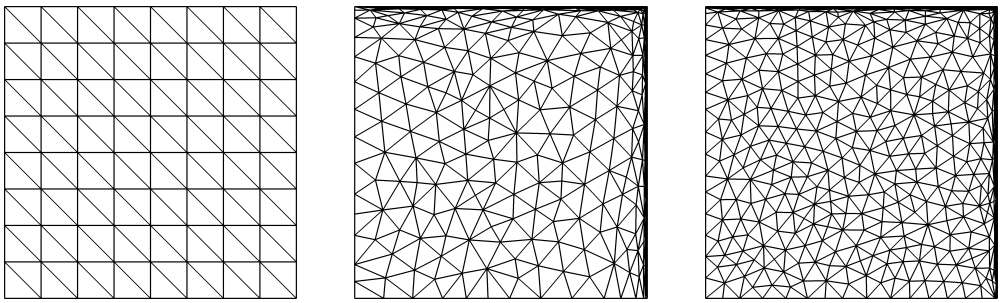
(a) Total number of iterations

(b) Error in H^1 seminorm and L^2 norm

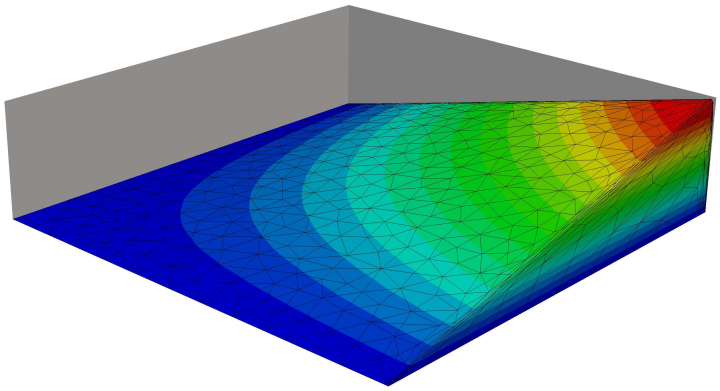
Figure 4.10: Results for the Laplace problem: each point represents one complete adaptive computation

using the GMRES algorithm, with ILU as a preconditioner.

In Figure 4.11 we show several adaptation steps from the initial mesh depicted in Figure 4.11(a) to the final mesh in Figure 4.11(c), and the contours of the computed solution on the final mesh in Figure 4.11(d).



(a) Initial mesh (b) 15th adptation step (c) Final mesh



(d) Final mesh and computed solution contours, 16 values between 0.0 and 0.9

Figure 4.11: Adaptation steps with final computed solution

In Table 4.2 we reported the results of the convergence analysis for four different adaptations with increasing refinement, where $N_{\mathcal{T}_h}$ is the number of vertices in the final mesh, and N_{alg} is the total number of linear solver iterations.

The drop in the number of linear iteration encountered for the Laplace problem is confirmed again in Figure 4.12(a) for the convection–diffusion problem.

In Figure 4.12(b) we assess that also for this test case we have the same level of accuracy between the compared results.

4.4.1.3 Convection-diffusion, double ramp

The last steady test case is a convection–diffusion problem solved on the L-shaped domain $\Omega = [(0,4) \times (0,4)] \setminus [(0,2) \times (0,2)]$ as proposed in [30, 31]. We assume diffusion coefficient $\kappa = 10^{-3}$, velocity field $\boldsymbol{v} = (1,0)^T$, homogeneous boundary

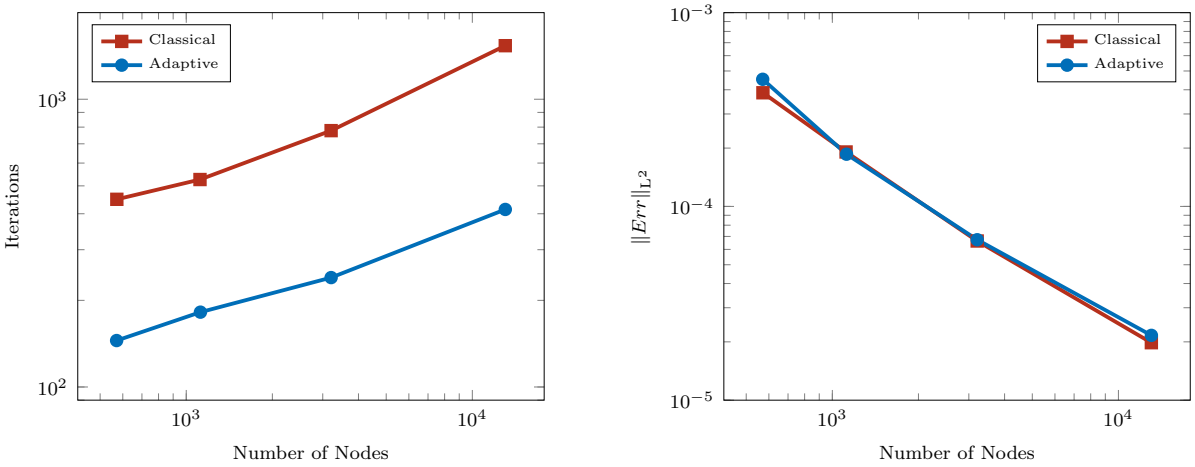
Table 4.2: Results for four different adaptations

(a) Classical stopping criterion

$N_{\mathcal{T}_h}$	N_{alg}	$\|u - u_h^n\|_{\mathrm{L}^2}$
572	449	$3.86 \cdot 10^{-4}$
1117	526	$1.91 \cdot 10^{-4}$
3209	778	$6.63 \cdot 10^{-5}$
13037	1535	$1.98 \cdot 10^{-5}$

(b) Adaptive stopping criterion

$N_{\mathcal{T}_h}$	N_{alg}	$\|u - u_h^n\|_{\mathrm{L}^2}$
571	145	$4.53 \cdot 10^{-4}$
1121	182	$1.86 \cdot 10^{-4}$
3210	240	$6.73 \cdot 10^{-5}$
13040	414	$2.16 \cdot 10^{-5}$



(a) Total number of iterations

(b) L^2 norm of the error

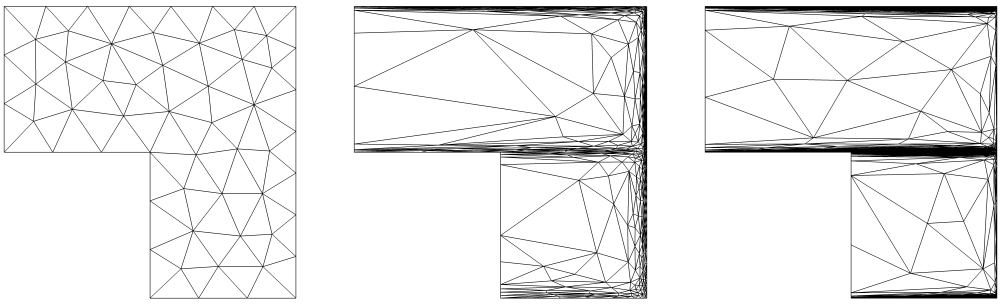
Figure 4.12: Results for convection–diffusion. Each point represents one complete adaptive computation

conditions $g = 0$ and source term $f = 1$.

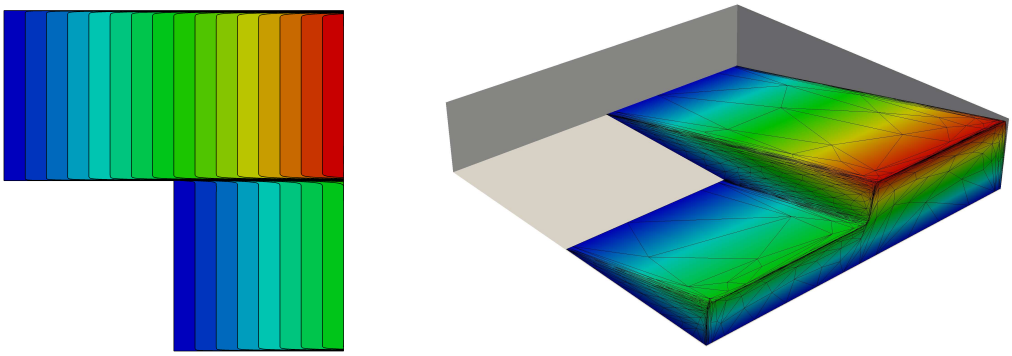
As in the previous test we use a GMRES solver with right ILU preconditioner.

In Figure 4.13 we show the initial mesh in Figure 4.13(a) and final mesh in Figure 4.13(c), and the contours of the computed solution in Figure 4.13(e) on the final adapted mesh.

We perform the same comparison using a classical stopping criterion and the new adaptive one. The results in Figure 4.14(a) indicate a significant drop in the number of iteration used by the GMRES solver.



(a) Initial mesh (b) 15th adptation step (c) Final mesh



(d) Iso-contours, 16 values (e) Final mesh and computed solution contours

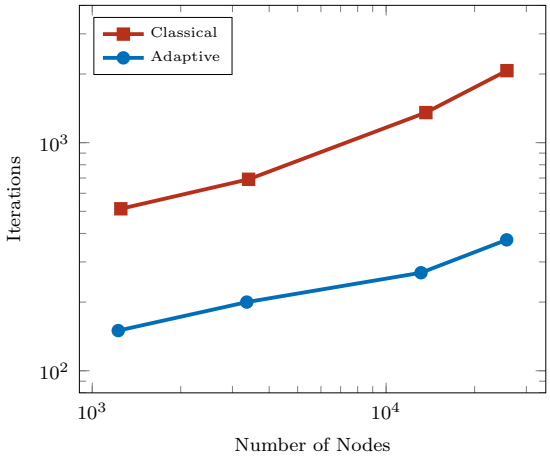
Figure 4.13: Adaptation steps with final computed solution, values between 0 and 4

4.4.1.4 Unsteady convection–diffusion, with internal layer

We present here an unsteady test case exhibiting evolving internal and boundary layers. The test case is taken from [32, 33],

$$\begin{cases} \partial_t u + \boldsymbol{v} \cdot \boldsymbol{\nabla} u - \kappa \boldsymbol{\Delta} u = f & \text{in } \Omega \times (0,T) \ , \\ u(\,.\,,0) = u_0 & \text{in } \Omega \ , \\ u = g & \text{on } \Gamma \ , \end{cases} \tag{4.32}$$

where we consider a constant velocity field $\boldsymbol{v} = (2,1)^T$ in the whole computational domain $\Omega = [0,1]^2$, a diffusion coefficient $\kappa = 10^{-3}$ and a zero source term. The



(a) Total number of iterations

Figure 4.14: Results for the double ramp. Each point represents one complete adaptive computation

initial condition is $u(x, y, 0) = 0$ except on $\partial\Omega$ where

$$u(x,y) = \begin{cases} 1 & if\ \{x = 0,\ 0 \leqslant y \leqslant 1\} \cup \{0 \leqslant x \leqslant 1,\ y = 1\}\ , \\ \dfrac{\delta - x}{\delta} & if\ \{x \leqslant \delta,\ y = 0\}\ , \\ \dfrac{y - 1 + \delta}{\delta} & if\ \{x = 1,\ y \geqslant 1 - \delta\}\ , \\ 0 & if\ \{x > \delta,\ y = 0\} \cup \{x = 1,\ y \leqslant 1 - \delta\}\ . \end{cases} \tag{4.33}$$

As time advances, the boundary layer at the left boundary propagates into the domain creating an internal layer that reaches the right wall resulting in a new boundary layer. The boundary layer at the top reduces progressively as the internal layer advances. The linear system that stems from the finite element discretization of this problem is solved using the GMRES algorithm, with ILU as a preconditioner.

For this test case we adopt a fixed time step $\Delta t = 0.001$ and a final time $T = 0.6$. The error estimator used for the stopping criterion can be either computed every mesh adaptation step and used in the following time steps without adaptation, or recomputed at each time step using the solution at previous step. If the solution is sufficiently regular the first choice allows reducing furthermore the required computational time, while the second choice is clearly more accurate. In this case we decide to compute the error estimator every time step, to better monitor its evolution.

In the first test, we adopt this strategy to apply the stopping criterion without mesh adaptation. We compute the solution on three fixed isotropic meshes of 5K, 10K and 50K elements, represented in Figure 4.15. Then we analyze the same

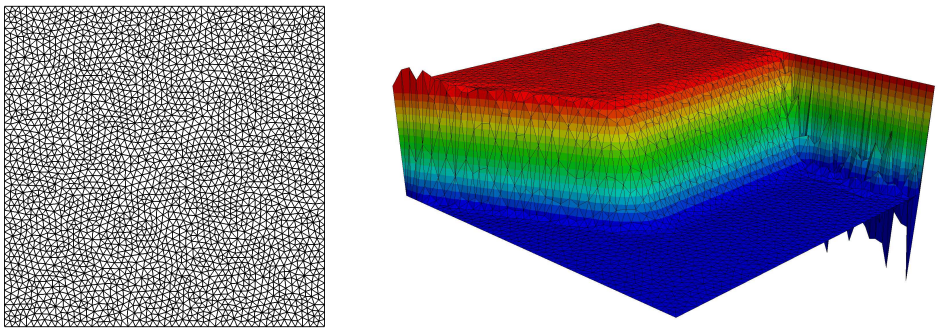


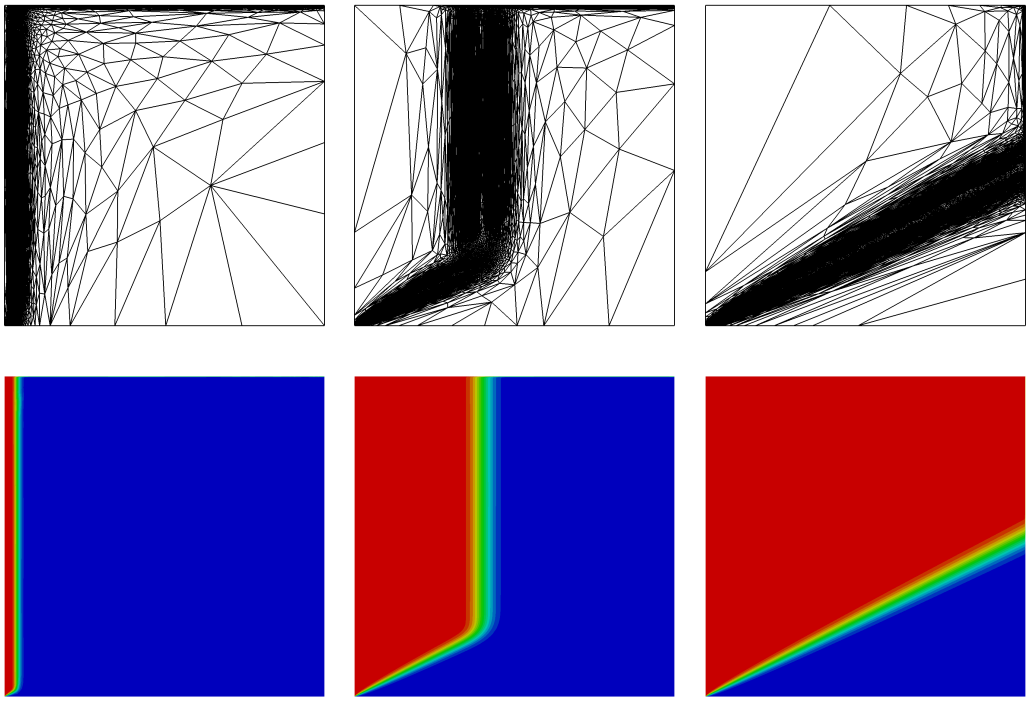
Figure 4.15: Mesh and solution contours at $t = 0.3$, 16 values between 0.0 and 1.0

test case using mesh adaptation with the same number of elements as a target. We perform a mesh adaptation step every 10 time steps. Due to the extremely anisotropic features, we obtain the highly stretched meshes shown in Figure 4.16, along with the solution contours, for the case with 5K elements at different time steps.

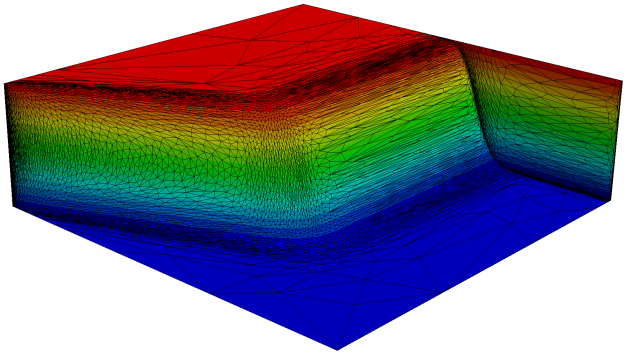
We compare the results obtained with a wide range of values for the constant c in the stopping criterion. We go from the more conservative value of $c = 0.01$, that imposes a reduction of the algebraic error to 1% of the estimated discretization error, to the value of $c = 1$, imposing the same level of error. This choice clearly affects the final gain in terms of number of iterations, but as we can see from the comparison in Figure 4.17 it does not affect the robustness of the method. Even if in some time steps we register some slightly differences in the isolines profile, mainly due to small discrepancies in the generated meshes, the accuracy of the solution is not spoiled at all. This remark is even more clear for the case without adaptation showed in Figure 4.18. We can also observe how the the use of highly stretched adapted meshes results in an increased solution accuracy, where with the same number of elements an isotropic mesh is unable to capture the boundary layers without oscillations.

In Figure 4.19(b) we compare the evolution of the discretization error estimator with the old and the new stopping criterion, and using different values of c. Figure 4.19(a) shows the final algebraic residual values for every time step, indicating the value at which the solver is stopped. We plot the results for the test with fixed mesh and with mesh adaptation. The result show that, except few oscillations, the estimated error is not affected by the use of the new stopping criterion, that does not spoil at all the accuracy of the solution.

Finally, we show in Figure 4.20(a) the total number of linear solver iterations needed for each complete simulation with the different settings, with and without mesh adaptation. We can see the considerable reduction achieved using the proposed



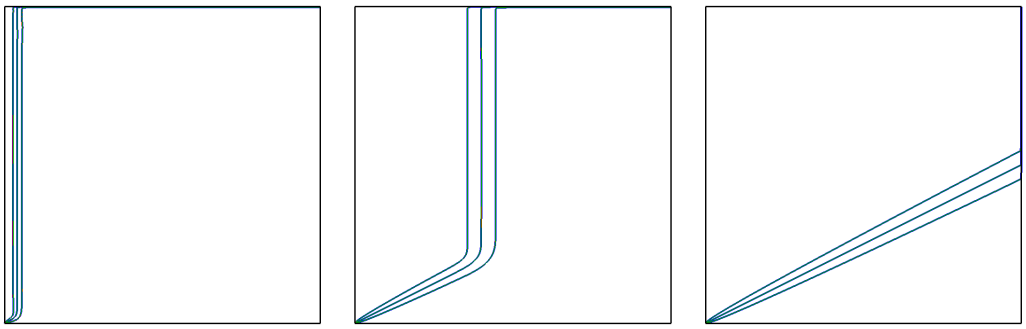
(d) $t = 0.02$ (e) $t = 0.2$ (f) $t = 0.6$



(g) Mesh and solution contours at $t = 0.3$

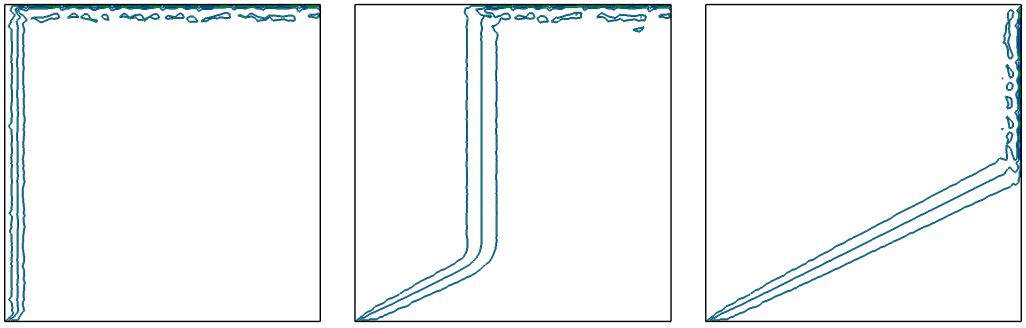
Figure 4.16: Adaptation steps and solution contours, 16 values between 0.0 and 1.0

adaptive stopping criterion. The evolution of the number of iterations for each time step is reported in Figure 4.20(b) for the test performed with mesh adaptation using a target elements number of 50K. We can point out how the use of the adaptive stopping criterion has the effect of leveling the number of iteration along the time



(a) $t = 0.02$ (b) $t = 0.2$ (c) $t = 0.6$

Figure 4.17: Isolines of the solution, values $\{0.1, 0.5, 0.9\}$. 5K elements adapted mesh. In blue the solution with classical stopping criterion, red $c = 0.01$, green $c = 1$.



(a) $t = 0.02$ (b) $t = 0.2$ (c) $t = 0.6$

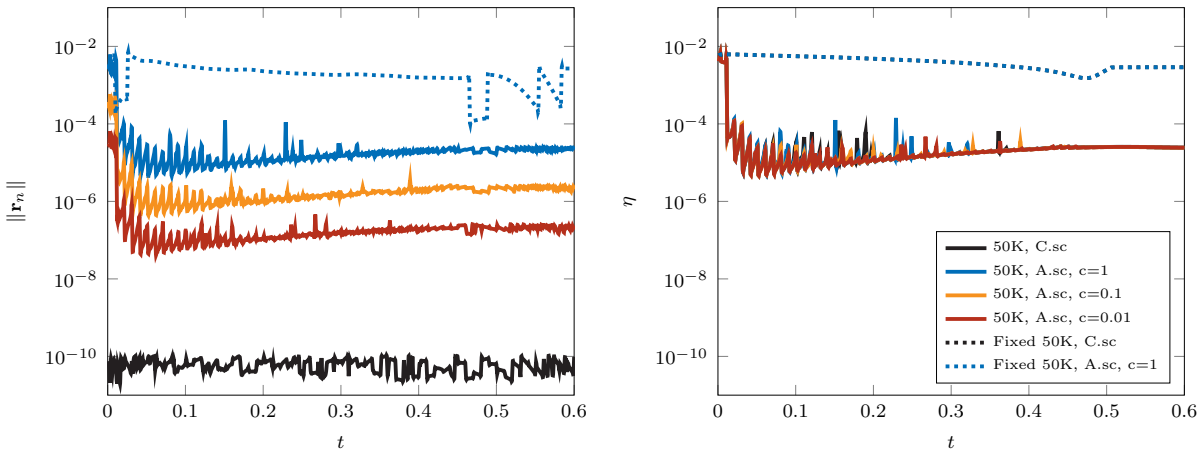
Figure 4.18: Isolines of the solution, values $\{0.1, 0.5, 0.9\}$. 5K elements fixed isotropic mesh. In blue the solution with classical stopping criterion, $c = 0.01$, green $c = 1$.

steps, avoiding useless extra accuracy, especially in the more demanding steps.

4.5 Adaptive stopping criterion for the Navier–Stokes equations

If we want to extend the proposed method to the iterative solution of the Navier–Stokes equations, we need to adjust some of the ingredients of the algorithm.

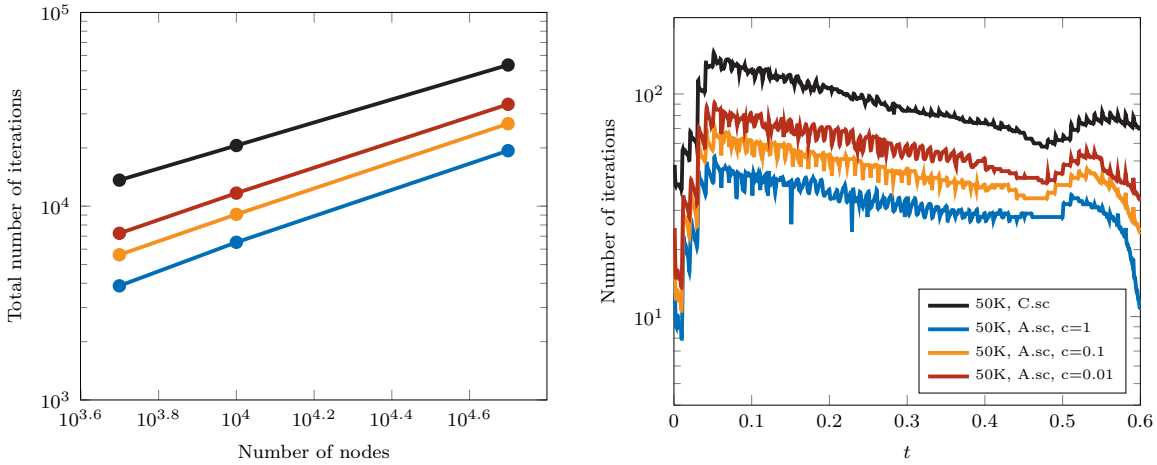
First of all we need a scalar function to be used as a function of interest for the interpolation error estimator. Velocity and pressure are the two variables in the equations, but the first is a vector field, that can not be feed as it is to either of the algorithms proposed above. Several possible choices are available; if we consider



(a) Final residual.

(b) Estimated error.

Figure 4.19: Evolution of final residual and estimated error. 50K elements mesh.



(a) Total number of iterations.

(b) Evolution of the number of iterations. 50K elements mesh.

Figure 4.20

the framework presented in Section 3.3.3, we could use a multi criteria technique to build a field function with several scalar fields of interest as proposed by Coupez in [34]. In this work we decided, for the sake of simplicity, to use the euclidean norm of the velocity that can be used indistinctly with both error estimators. We consider, then, $u_h = \|\boldsymbol{v}_h\|$ the finite element approximation of scalar field $u = \|\boldsymbol{v}\|$, where $\boldsymbol{v}_h$ is the finite element approximation of the exact velocity solution $\boldsymbol{v}_h$ from the Navier–Stokes equation. We denote u_h^n the approximation of u_h from the iterative solution, at the n_{th} iteration.

Secondly we need to consider that the finite element system coming from the discretization of the Navier–Stokes equations is non-linear. Several iterative algorithms are available to solve this type of system, e.g. fixed point methods and Newton methods [35]. In terms of stopping criteria, this means that there will be one stopping criterion to control the outer non-linear iterations and one stopping criterion to control the inner linear iterations. Comparably with the linear solvers, in the literature several approached are proposed to control the non-linear convergence. Generally one can fix a given number of iterations at which to stop (as for Inexact Newton methods) or continue iterating until the non-linear residual is very small. If we had to follow the adaptive procedure presented for the linear case, we could apply recursively the same idea. Then the non-linear iteration would be stopped when the linearization error η_{lin} is smaller than the interpolation error, and the linear iterations when the algebraic error is smaller then the linearization one

$$\eta_{alg} \ll \eta_{lin} \ll \eta \ . \tag{4.34}$$

This would result in a recursive application of the same adaptive stopping procedure. When we consider industrial cases, however, using non linear solvers in a proper sense is often too computationally expensive. A typical approach is to use a fixed point algorithm with only one outer iteration. This results in the actual solution of the inner linear system only. Due to the aim of this work to be applied at the solution of complex industrial simulations, we decide to apply this more simple approach to evaluate the effect of our proposed stopping criterion to reduce the required computational time.

The adaptive stopping criterion used here is then equivalent to the one in Equation (4.25) for the CDR equation, extended to be used for the solution Navier–Stokes equation

$$\|r(\boldsymbol{v})^n\| \leqslant c \left(\sum_{K \in \mathcal{T}_h} \eta(\boldsymbol{v})_K^2 \right)^{1/2} \ . \tag{4.35}$$

where $\|r(\boldsymbol{v})^n\|$ is the algebraic residual related to the velocity, and $\eta(\boldsymbol{v})$ is the estimator computed with the field function mentioned above.

In the following section we validate the proposed stopping criterion with the popular benchmark of the 2D flow past a square cylinder.

4.5.1 2D laminar flow past a square cylinder

To validate the proposed method, we choose the well known square cylinder test case, treated among others by [36–39]. A two-dimensional square cylinder is placed in a computational domain, with his center in the origin of the coordinate system. The cylinder is exposed to a constant free-stream velocity U. As shown in Figure 4.21, the

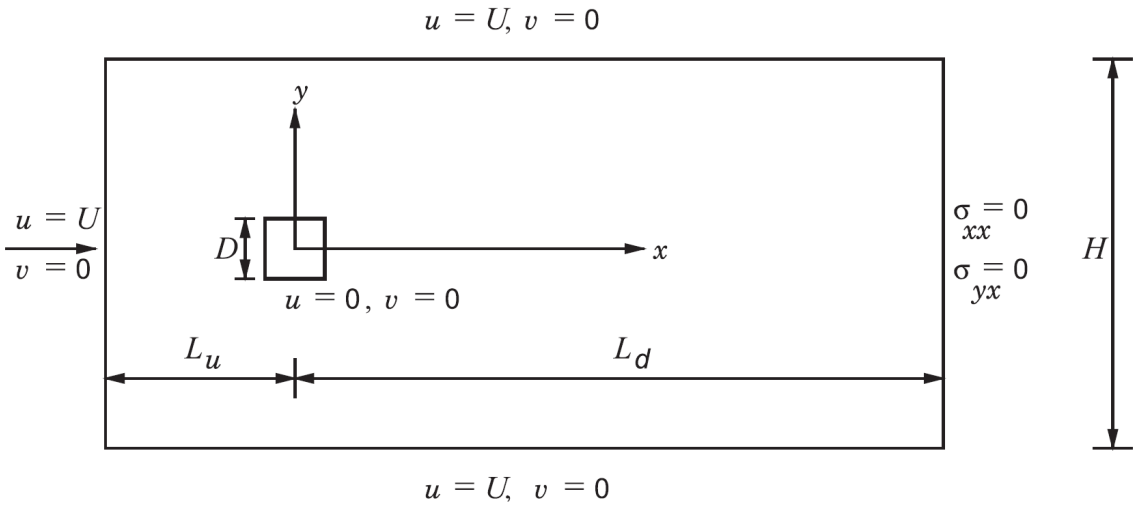


Figure 4.21: Problem definition

square has size D and the distances from the upstream and downstream boundaries are respectively L_u and L_d, H is the distance between the sidewalls. Sohankar et al. [40] studied the effects of the placing of the boundaries on the flow. Following their conclusions we consider D as the nondimensionalizing length scale and we choose $H = 20$ for a blockage $b \leq 5\%$, and we increased the horizontal lengths to $L_u = 10D$, $L_d = 30D$ as a safety measure.

4.5.1.1 The finite elements mesh

For the simulations performed on a fix mesh, we discretize the computational domain with an isotropic mesh of triangles. We refined the mesh around the cylinder, as shown in Figure 4.22. To validate the choice of the mesh, we perform the simulation on five meshes M1, M2, M3, M4, M5. Each mesh is obtained from the previous one applying a global uniform refinement, reducing the element size by 25%. The simulations performed with the mesh adaptation technique presented in Section 3.2, are carried out using an initial coarse mesh and performing one adaptation step every three time steps. An example of one resulting adapted meshes is shown in Figure 4.23, where the two snapshots (b) and (c) are taken with an interval of roughly one-half of the shedding period.

The details of the meshes and the obtained results are listed in Table 4.3, in Figure 4.24 we show the convergence on the values of drag and lift coefficients (Cd, Cl).

We can observe that the step of refinement from M4 to M5 has a negligible effect of the results, so we will use M4 for all the following tests. The comparison of the integral flow parameters obtained on the M4 mesh with earlier results available in

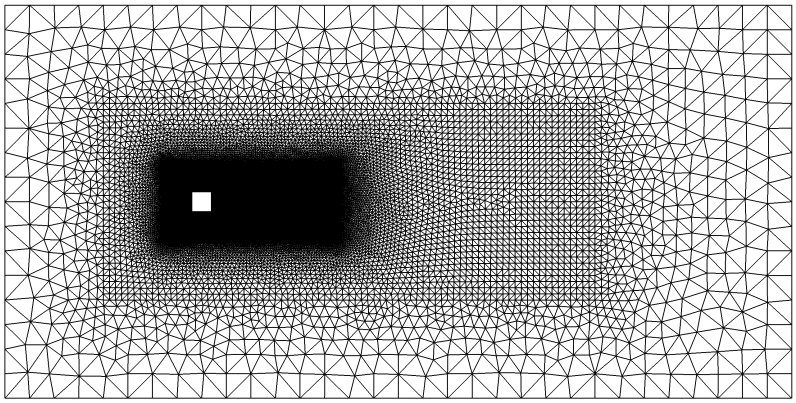
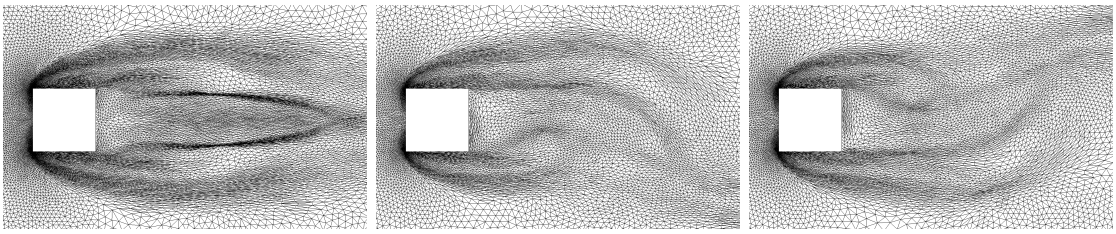


Figure 4.22: Isotropic triangular mesh M1.

Table 4.3: Details of the different meshes used for the convergence study and flow parameters at Re=100.

Mesh	Elements	Cd_{avg}	Cl_{rms}
M1	35892	1.5034	0.2034
M2	50968	1.4989	0.2011
M3	79270	1.4966	0.2006
M4	142202	1.4954	0.2004
M5	315336	1.4952	0.2002
Adapted[1]	30000	1.4947	0.1999



(a) (b) (c)

Figure 4.23: Adapted meshes, zoom on the object proximity.

the literature is presented in table 4.4. The time averaged drag coefficient (Cd_{avg}) is in good agreement with the references and the Strouhal number ($St = fD/U$, where f is the frequency of shedding) is within 3%. The r.m.s. value of the lift coefficient (Cl_{rms}) is slightly higher than the other references, nonetheless it is within 4% from

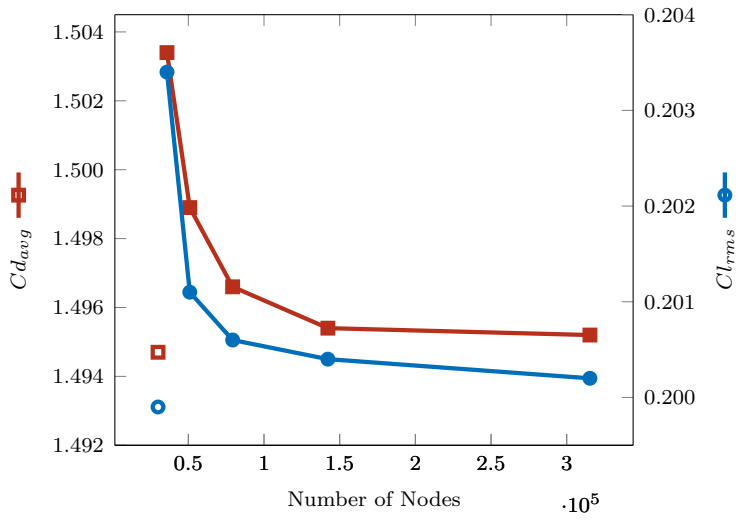


Figure 4.24: Mesh convergence results, Re=100. Solid lines for the simulations using a fix mesh, empty markers for the simulation with mesh adaptation.

the one reported by Sen et al. [36]. Figure 4.25 shows the contours of velocity with

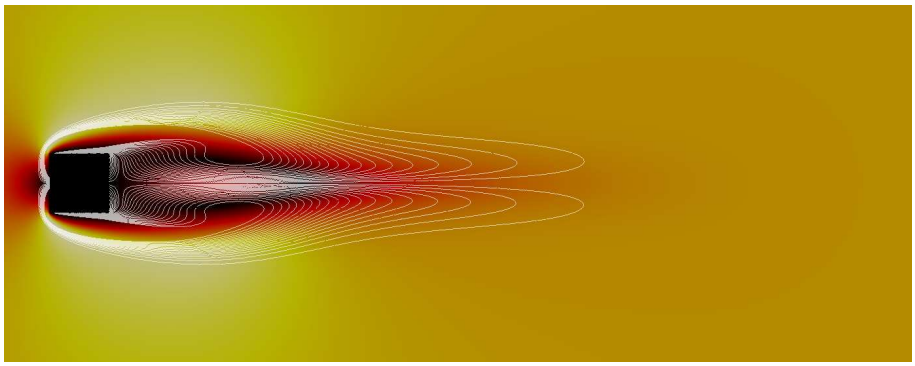
Table 4.4: Comparison of integral flow parameters with references at Re=100.

Reference	B	Cd_{avg}	Cl_{rms}	St
Sharma and Eswaran [38]	0.0500	1.4936	0.1922	0.1488
Darekar and Sherwin [39] (3D)	0.0230	1.4860	0.1860	0.1460
Sahu et al. [37]	0.0500	1.4878	0.1880	0.1486
Sen et al. [36]	0.0500	1.5287	0.1928	0.1452
Present (Fix)	0.0500	1.4966	0.2006	0.1416
Present (Adapted)	0.0500	1.4947	0.1999	0.1416

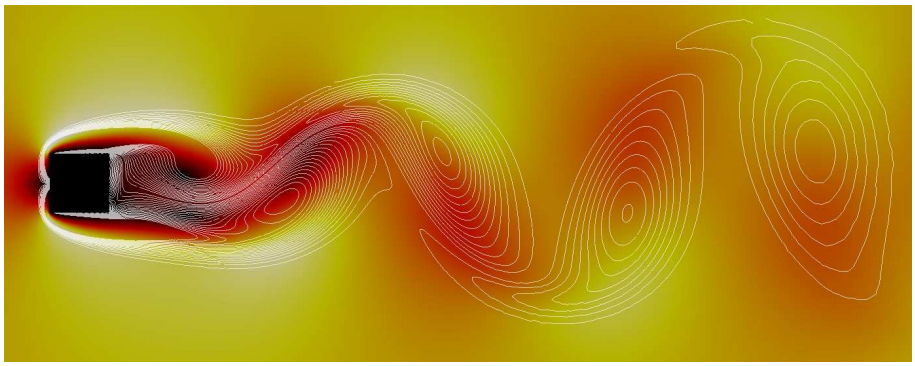
isovalues of vorticity, at different time steps, obtained on the M4 fix mesh. In (a) the solution has yet to develop the instability that is highlighted in (b) and (c), where we can see the typical Karman vortex street; the two snapshots are taken with an interval of roughly one-half of the shedding period.

4.5.1.2 Stopping criteria comparison. Unsteady flow at Re=100.

We present here the validation of the method proposed in Section 4.3 on the unsteady flow at Re=100 past a stationary square cylinder test case, presented above. We carried out the simulations on a fix mesh and using mesh adaptation, comparing the



(a)



(b)

(c)

Figure 4.25: Velocity contours {0,1.4}, with isovalues of vorticity

proposed adaptive stopping criterion:

$$\|\mathbf{r}_n\| \leqslant c \left(\sum_K \eta_K^2 \right)^{1/2} , \tag{4.36}$$

with a classical stopping criterion, where we impose a given precision (e.g., $\epsilon = 10^{-6}$)

$$\|\mathbf{r}_n\| \leq 10^{-6}. \tag{4.37}$$

We point out here the given precision we choose here is not extremely strict (far from machine precision) but is typically used by the final users of commercial software, although no theory is behind this choice, but only the specific experience of the specific user.

In Table 4.5 we provide the results of the comparison in terms of precision and computational time needed depending on the stopping criterion used – C (classical) A (adaptive) – and the scaling factor for the stopping criterion, from Equation (4.25). The results show that the application of the proposed stopping criterion for the

Table 4.5: Comparison of integral flow parameters with references at Re=100.

Mesh	Criterion	Scaling	Cd_{avg}	Cl_{rms}	Walltime (min)	
M4	C	/	1.4966	0.2006	84.9	
M4	A	c=0.1	1.4955	0.2004	39.5	-65%
M4	A	c=1.0	1.4955	0.2004	15.2	-82%
Adapted	C	/	1.4948	0.1999	10.3	
Adapted	A	c=0.1	1.4946	0.1997	4.8	-53%
Adapted	A	c=1.0	1.4937	0.1996	2.1	-79%

iterative solver does not affect the precision of the measured integral values, being the shift always below 0.1%. On the other side we highlight a great impact on the computational time needed for the iterative solution, where we find a reduction up to 82% compared to a classical stopping criterion. The estimated error in L^2-norm is in the order of magnitude of 10^{-4} for both fix and adapted mesh simulations.

4.6 Towards industrial applications: 3D fluid flow with thermal coupling

Heat transfer is involved in several physical processes, and it can be the limiting factor for many of them. The modeling of heat transfer effects inside industrial furnaces as depicted in Figure 4.26 is nowadays a developing field of investigation as a result of the demand for energy conservation through efficiency improvement and for reduction of pollutant emissions. It also more and more important in the design of the products itself in many areas such as the electronics, automotive, machinery and equipment manufacturing industries. A heat treatment in a furnace is a

(a) (b)

Figure 4.26: Examples of industrial furnaces

manufacturing process to control the mechanical and physical properties of metallic components. In the design of this process furnace control, turbulent flows, conduction within the load, convection and thermal radiation are involved simultaneously. The thermal history of each part and the temperature distribution in the whole load are critical for the final microstructure and the mechanical properties of workpieces and can directly determined the final quality of parts in terms of hardness, toughness and resistance. To achieve higher treatment efficiency, the major influencing factors such as the design of the furnace, the location of the workpieces, thermal schedule and position of the burners should be understood thoroughly. The major factor to be considered in the working of a furnace is the heat transfer by all the modes, which occur simultaneously. To either study a new furnace or to optimize the heating process in existing ones, the heat transfer in the furnace has to be modeled in the same way of a real situation as closely as possible. Given the geometry of the furnace, different boundary conditions along the furnace length, gas composition and properties and other complexities, an analytical solution in not feasible and computational modeling has to be resorted to. A CFD model of the heating process should be capable of doing so in an accurate way and within a reasonable time. Modeling of heat transfer for heat treatment furnaces should describe convection, radiation, turbulent flow and furnace control. Conduction mainly occurs in all solids materials. Turbulent convection exists between the atmosphere and solid materials exposed to it, and furnace walls to the ambient air; in our case the effects of turbulence are taken into account by means of the $k - \omega$ model for the sake of efficiency. Radiation exists between solid materials exposed to each other and to all walls.

We present here a 3D test case devised to model the fluid flow with heat transfer

inside an industrial furnace.

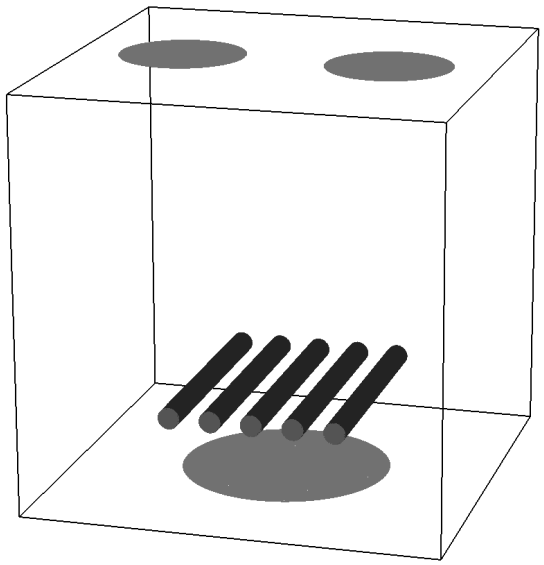
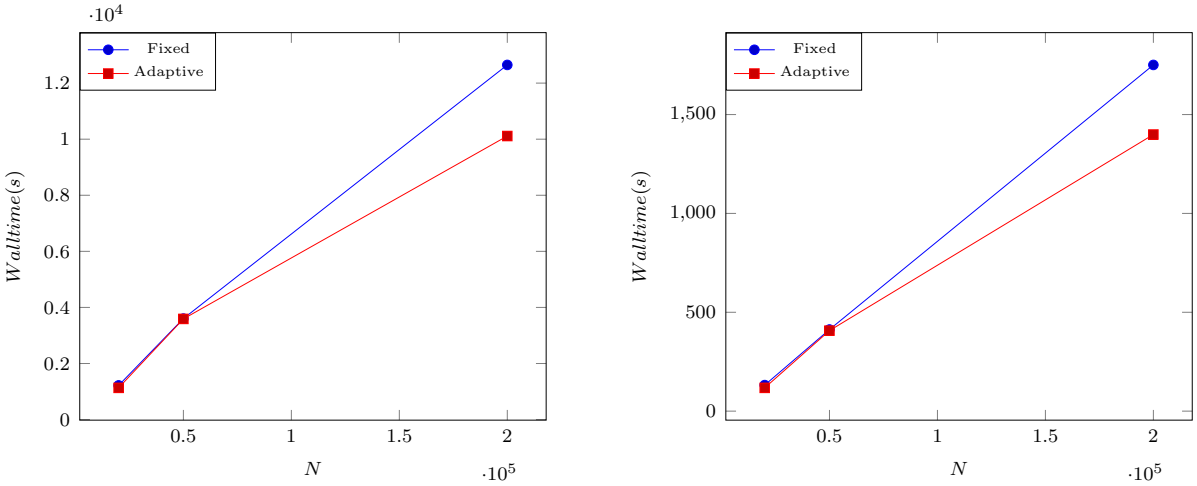


Figure 4.27: Problem definition

The Navier–Stokes Equations (2.24) are coupled with the thermal convection–diffusion–reaction Equation (2.1). As shown in Figure 4.27, we have a cubic domain with one inlet on the lower wall and two outlets on the top wall. Inside the cube we placed five cylinders to be subjected to the thermal treatment. The air is introduced into the furnace at a constant velocity $U = 0.5m/s$ with a temperature of $400K$. On all the walls we impose and adiabatic condition for the temperature. We advance the simulation to $t = 150s$, with a time step $dt = 0.01s$.

As a preliminary study we perform a few timings on the simulation runs for meshes containing a moderate number of vertices: 20K, 50K, and 200K. This problem dimension is not enough for an accurate simulation of such thermo-mechanical system but it is enough to demonstrate a sensible performance increase as depicted in Figures 4.28(a) and 4.28(b), which corresponds to a reduction by 20 percent of the simulation walltime.

The main study consists of comparing the resolution times for a full-scale resolved simulation and evaluating the influence of the stopping criterion on the quality of the discrete solution. We perform first a comparison on a fixed mesh made of 3.5M tetrahedral elements. In Figure 4.29 we show the contours of temperature for several time steps. We compare the time averaged values of temperature on a diagonal line that lays on the horizontal plane over the cylinders, using a classical or an adaptive stopping criterion. The plot in Figure 4.30 shows that the difference between the two results are negligible, and the proposed method does not reduce the accuracy



(a) Navier–Stokes (b) CDR

Figure 4.28: Resolution time for the solvers of the 3D fluid flow with thermal coupling

of the solution.

In Table 4.6 we provide the results of the comparison of the stopping criteria in terms of computational time needed: The results show a great impact on the

Table 4.6: Comparison of integral flow parameters with references at Re=100.

Mesh	SC	Scaling	CPUsol(h)	
Fix 3.5M	C	/	80.6	
Fix 3.5M	A	c=0.01	47.4	-41%

computational time needed for the iterative solution, with a reduction up to 41% compared to a classical stopping criterion on a realistic simulation with a large number of degrees of freedom.

4.7 A note on the preconditioner side

In all the test analysed above we assumed the use of a preconditioning technique applied to the iterative solver. In the case of GMRES we have to make a distinction between the two possible choices for the preconditioner side.

On one side, if we apply a left preconditioner P_l to the linear system $\mathrm{A}\mathbf{x} = \mathbf{b}$ we have the following system to solve

$$\mathrm{P}_l^{-1}\mathrm{A}\mathbf{x} = \mathrm{P}_l^{-1}\mathbf{r} \ , \tag{4.38}$$

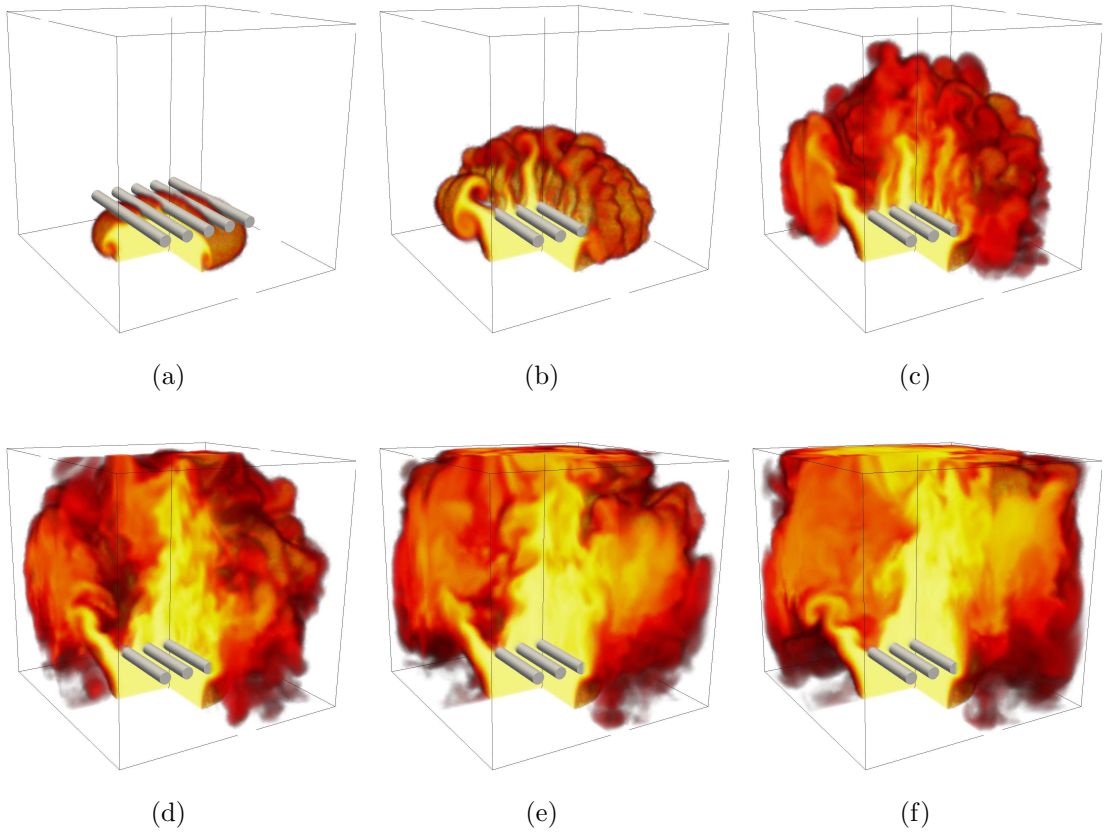


Figure 4.29: Temperature contours with one quarter section removed, values from $300K$ *to* $400K$

and the convergence is controlled monitoring the residual of this linear system $\mathbf{r}^* = \mathrm{P}_l^{-1}\mathbf{r}$, that is the preconditioned residual of the original linear system. On the other side, if we choose a right preconditioner P_r we have

$$(\mathrm{A}\mathrm{P}_r^{-1})(\mathrm{P}_r\mathbf{x}) = \mathbf{r} \ . \tag{4.39}$$

In this case the residual is not affected by the application of the preconditioner, and we can directly use it to control the convergence.

In the following we present a comparison between these two preconditioning techniques applied to the ILU(k) preconditioner. Due to the strong anisotropy of our adapted meshes we decide to use a fill-in k=1 to have a better rate of convergence. In Figure 4.31 we show the values of the residual norm for a complete iterative solution of the convection–diffusion test case, analysed in 4.4.1.2, on the final mesh of the first refinement level (558 nodes). The plots show that the use of a right preconditioner reveals a difficulty damping the higher frequencies of the error, resulting in

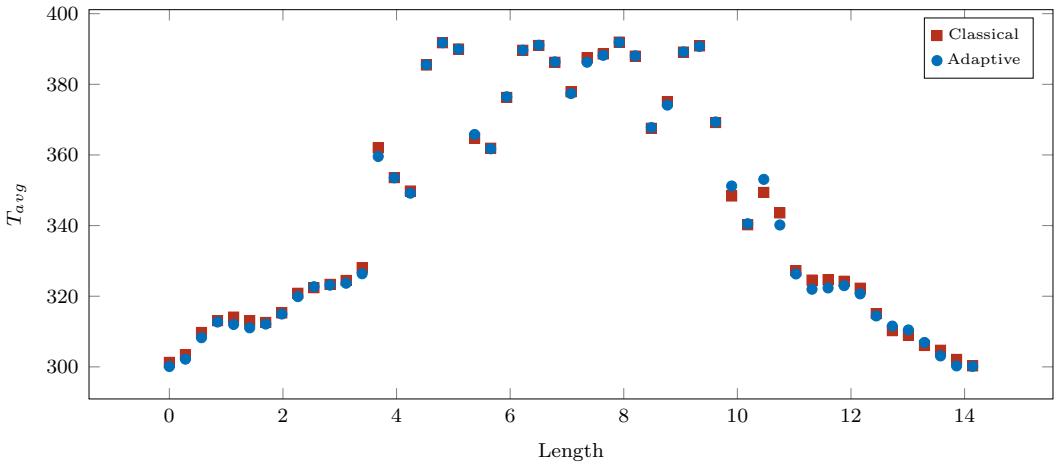


Figure 4.30: Time averaged temperature on the diagonal above the cylinders.

an initial plateau. This will lead to poor convergence especially if we use an inadequate preconditioner, or if the GMRES algorithm is restarted after few iterations as depicted in Figure 4.31(b) before the asymptotic convergence behaviour is realized. The use of a left preconditioner hides this behaviour using r^*, that is amplified by P_r, and increases the convergence rate in this first part. However this will result in a poor control of the value of the residual norm, and possibly to a higher number of iterations used, Figure 4.31(a).

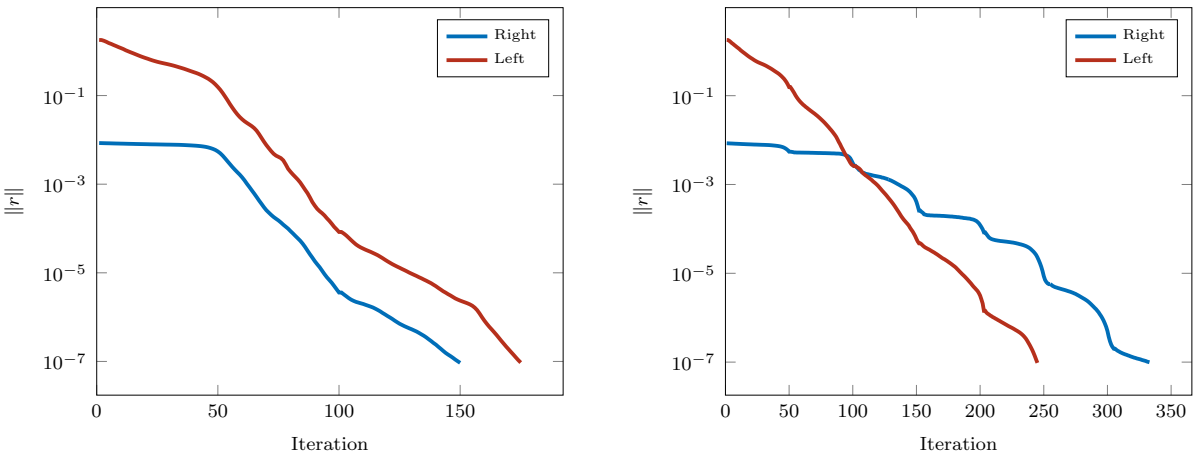
To relate this analysis with the subject of this paper, we refer again to the test in 4.4.1.2. On the first refinement level with the adaptive stopping criterion, the use of a left preconditioner leads to doubling the number of iteration needed (from 58 to 126).

In conclusion the use of a right preconditioner is mandatory to ensure the reliable control of the convergence needed by an adaptive stopping criterion.

4.8 Conclusions

In this chapter we proposed an automatic adaptive stopping criterion for iterative solvers in the framework of anisotropic adapted finite elements, applied to convection-dominated problems. The proposed formulation takes advantage from the information computed in the mesh adaptation procedure, that provides an estimate of the approximation error with no additional computational cost. We propose to stop the iterative solver when the algebraic error is below the level of the estimated approximation error.

Numerical tests have been performed for a benchmark symmetric Laplace problem, several steady and unsteady convection–diffusion, and Navier–Stokes problems



(a) GMRES(100)

(b) GMRES(50)

Figure 4.31: GMRES convergence

solved with stabilized finite elements on highly stretched meshes. These test cases have been chosen for their straightforward implementation, to be used as future benchmarks on the subject. The results proved that, following the provided approach, the total number of iterations needed can be reduced up to five times, with no significant effect on the accuracy of the computed solution. This results in a considerable reduction of the computational time dedicated to the iterative solution of the problem.

Finally a model problem of heat treatment in a furnace was investigated to demonstrate the applicability of the developed approach to realistic full-scale simulations. The numerical solver coupling Navier–Stokes and the heat transfer equation was run with adaptive stopping criteria based on two a posteriori error estimators. The computational time was shown to be reduced favorably while a good quality of the discrete solution with respect to the quantities of interest was maintained: this is promising for improving the performance of simulations of industrial applications in engineering.

4.9 Bibliography

[1] R. Becker, C. Johnson, R. Rannacher, Adaptive error control for multigrid finite element, Computing 55 (4) (1995) 271–288. 64

[2] M. Arioli, A stopping criterion for the conjugate gradient algorithm in a finite element method framework, Numerische Mathematik 97 (1) (2004) 1–24. v, 65, 71, 73

[3] M. Picasso, A stopping criterion for the conjugate gradient algorithm in the framework of anisotropic adaptive finite elements, Communications in Numerical Methods in Engineering 25 (4) (2009) 339–355. 65, 72, 76, 79

[4] A. Ern, M. Vohralík, Adaptive inexact newton methods with a posteriori stopping criteria for nonlinear diffusion PDEs, SIAM Journal on Scientific Computing 35 (4) (2013) A1761–A1791. 65, 74

[5] P. Jiránek, Z. Strakoš, M. Vohralík, A posteriori error estimates including algebraic error and stopping criteria for iterative solvers, SIAM Journal on Scientific Computing 32 (3) (2010) 1567–1590. 65, 75

[6] R. Stevenson, Optimality of a standard adaptive finite element method, Foundations of Computational Mathematics 7 (2) (2007) 245–269. 65

[7] M. Arioli, J. Liesen, A. Miçdlar, Z. Strakoš, Interplay between discretization and algebraic computation in adaptive numerical solution of elliptic PDE problems, GAMM-Mitteilungen 36 (1) (2013) 102–129. 65

[8] G. Manzinali, E. Hachem, Y. Mesri, Adaptive stopping criterion for iterative linear solvers combined with anisotropic mesh adaptation, application to convection-dominated problems, Computer Methods in Applied Mechanics and Engineering 340 (2018) 864–880. 65

[9] L. Kamenski, W. Huang, A study on the conditioning of finite element equations with arbitrary anisotropic meshes via a density function approach, Journal of Mathematical Study 14 (2014) 151–172. 66

[10] L. Kamenski, W. Huang, H. Xu, Conditioning of finite element equations with arbitrary anisotropic meshes, Mathematics of Computation 83 (289) (2014) 2187–2211. 66

[11] R. E. Bank, L. R. Scott, On the conditioning of finite element equations with highly refined meshes, SIAM Journal on Numerical Analysis 26 (6) (1989) 1383–1394. 66

[12] Y. Saad, Iterative methods for sparse linear systems, Vol. 82, siam, 2003. 68

[13] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van der Vorst, Templates for the solution of linear systems: building blocks for iterative methods, Vol. 43, Siam, 1994. 68

[14] E. Cuthill, Several strategies for reducing the bandwidth of matrices, in: Sparse matrices and their applications, Springer, 1972, pp. 157–166. 70

[15] P. R. Amestoy, T. A. Davis, I. S. Duff, An approximate minimum degree ordering algorithm, SIAM Journal on Matrix Analysis and Applications 17 (4) (1996) 886–905. 70

[16] A. George, Nested dissection of a regular finite element mesh, SIAM Journal on Numerical Analysis 10 (2) (1973) 345–363. 70

[17] L. C. Dutto, The effect of ordering on preconditioned GMRES algorithm, for solving the compressible navier-stokes equations, International Journal for Numerical Methods in Engineering 36 (3) (1993) 457–497. 70

[18] M. Benzi, D. B. Szyld, A. Van Duin, Orderings for incomplete factorization preconditioning of nonsymmetric problems, SIAM Journal on Scientific Computing 20 (5) (1999) 1652–1670. 70

[19] J. Camata, A. Rossa, A. Valli, L. Catabriga, G. Carey, A. Coutinho, Reordering and incomplete preconditioning in serial and parallel adaptive mesh refinement and coarsening flow solutions, International Journal for Numerical Methods in Fluids 69 (4) (2012) 802–823. 70

[20] J. Bey, G. Wittum, Downwind numbering: Robust multigrid for convection-diffusion problems, Applied Numerical Mathematics 23 (1) (1997) 177–192. 70

[21] G. H. Golub, G. Meurant, Matrices, moments and quadrature ii; how to compute the norm of the error in iterative methods, BIT Numerical Mathematics 37 (3) (1997) 687–705. 72

[22] M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems 49 (1952). 72

[23] M. Picasso, An anisotropic error indicator based on zienkiewicz–zhu error estimator: Application to elliptic and parabolic problems, SIAM Journal on Scientific Computing 24 (4) (2003) 1328–1355. 73

[24] O. C. Zienkiewicz, J. Z. Zhu, A simple error estimator and adaptive procedure for practical engineerng analysis, International journal for numerical methods in engineering 24 (2) (1987) 337–357. 73

[25] W. Prager, J. L. Synge, Approximations in elasticity based on the concept of function space, Quarterly of Applied Mathematics 5 (3) (1947) 241–269. 75

[26] M. Arioli, E. H. Georgoulis, D. Loghin, Stopping criteria for adaptive finite element solvers, SIAM Journal on Scientific Computing 35 (3) (2013) A1537–A1559. 76

[27] L. Formaggia, S. Perotto, Anisotropic error estimates for elliptic problems, Numerische Mathematik 94 (1) (2003) 67–92. 79

[28] H. Nguyen, M. Gunzburger, L. Ju, J. Burkardt, Adaptive anisotropic meshing for steady convection-dominated problems, Computer Methods in Applied Mechanics and Engineering 198 (37) (2009) 2964 – 2981. 80

[29] Z. Zhang, Finite element superconvergence on shishkin mesh for 2-d convection-diffusion problems, Mathematics of computation 72 (243) (2003) 1147–1177. 80

[30] L. Formaggia, S. Micheletti, S. Perotto, Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the stokes problems, Applied Numerical Mathematics 51 (4) (2004) 511 – 533, applied Scientific Computing: Advances in Grid Generatuion, Approximation and Numerical Modeling. 82

[31] C. Farhat, I. Kalashnikova, R. Tezaur, A higher-order discontinuous enrichment method for the solution of high Péclet advection-diffusion problems on unstructured meshes, International Journal for Numerical Methods in Engineering 81 (5) (2010) 604–636. 82

[32] G. Jannoun, E. Hachem, J. Veysset, T. Coupez, Anisotropic meshing with time-stepping control for unsteady convection-dominated problems, Applied Mathematical Modelling 39 (7) (2015) 1899 – 1916. 84

[33] M. Picasso, V. Prachittham, An adaptive algorithm for the Crank-Nicolson scheme applied to a time-dependent convection-diffusion problem, Journal of Computational and Applied Mathematics 233 (4) (2009) 1139 – 1154. 84

[34] T. Coupez, E. Hachem, Solution of high-reynolds incompressible flow with stabilized finite element and adaptive anisotropic meshing, Computer Methods in Applied Mechanics and Engineering 267 (2013) 65–85. 89

[35] C. T. Kelley, Iterative methods for linear and nonlinear equations, Vol. 16, SIAM, 1995. 90

[36] S. Sen, S. Mittal, G. Biswas, Flow past a square cylinder at low Reynolds numbers, International Journal for Numerical Methods in Fluids 67 (9) (2011) 1160–1174. 90, 93

[37] A. K. Sahu, R. Chhabra, V. Eswaran, Two-dimensional unsteady laminar flow of a power law fluid across a square cylinder, Journal of Non-Newtonian Fluid Mechanics 160 (2-3) (2009) 157–167. 93

[38] A. Sharma, V. Eswaran, Heat and fluid flow across a square cylinder in the two-dimensional laminar flow regime, Numerical Heat Transfer, Part A: Applications 45 (3) (2004) 247–269. 93

[39] R. M. Darekar, S. J. Sherwin, Flow past a square-section cylinder with a wavy stagnation face, Journal of Fluid Mechanics 426 (2001) 263–295. 90, 93

[40] A. Sohankar, C. Norberg, L. Davidson, Low-Reynolds-number flow around a square cylinder at incidence: study of blockage, onset of vortex shedding and outlet boundary condition, International journal for numerical methods in fluids 26 (1) (1998) 39–56. 91

Chapter 5

Conclusions and Perspectives

Résumé *La formulation proposée tire parti des informations calculées dans la procédure d'adaptation du maillage, qui fournit une estimation de l'erreur de discrétisation sans coût de calcul supplémentaire. Nous montrons que le solveur itératif peut être arrêté lorsque l'erreur algébrique est inférieure au niveau de l'erreur de discrétisation estimée.*

5.1 Conclusions

The objective of this thesis is the development of a first, practical and general stopping criterion using an a posteriori approach, that relies on the error estimates available from the mesh adaptation procedure. This stopping criterion must be robust and applicable to the different types of equations used to describe the complex physics involved in a conjugate heat transfer problem. The final goal is to prove that with such stopping criterion it is possible to drastically reduce the CPU time required for the solution of coupled systems.

Such system involves incompressible flows with conjugate heat transfer all in a multi-component formulations. The tools used in this thesis are the Finite Element Method (FEM) and Computational Fluid Dynamics (CFD). This method is shown as an attractive way to solve the turbulent flow and heat transfer and it can be applied for a variety of geometry and boundary conditions.

Therefore, the first part of the thesis is centered on the modeling and solution methods for the heat transfer and fluid flows problems. Standard finite element method normally exhibits global spurious oscillations in convection-dominated problems, especially in the vicinity of sharp gradients. To overcome this limitation, we chose the Streamline Upwind Petrov–Galerkin (SUPG) method applied with convection-dominated convection–diffusion–reaction problems. This approach is efficient to eliminate the instabilities adding a numerical diffusion in the convective direction.

To solve the fluid flow problem, we use a variational multiscale approach to solve the incompressible Navier-–Stokes equations. This stabilization method is proven to be efficient in handling convection-dominated problems and deal with the inf-sup stability condition. When dealing with turbulent flows, we introduced the RANS equations with turbulence models, that provide the averaged fluid flow solution without solving the turbulence small scales. The numerical experiments show that the coupled system is stable. The performance and the efficiency of the overall scheme have been demonstrated by two benchmarks.

Following the solver part, we introduce the metric-based anisotropic mesh adaptation used to provide an optimal mesh and very useful to capture steep gradients in the fields of interest. We outline the principles of the metric-based approach, to build a metric tensor in the Riemann space. The result of the procedure is a set of stretching factors that are used to adapt the mesh, resulting in an anisotropic mesh that is well adapted on the field of interest. The adaptation procedure is driven by an error estimator. We propose two a-posterior error estimators, one based on the recovered Hessian of the solution, the second relying on the recovered gradients on the edge of the elements. Both estimators are general and PDE-independent, and can be applied to any field of interest.

While combining both previous part, we complete the manuscript by proposing an automatic adaptive stopping criterion for iterative linear solvers in the framework of anisotropic adapted finite elements, applied to symmetric and non-symmetric problems. The proposed formulation takes advantage from the information computed in the mesh adaptation procedure, that provides an estimate of the discretization error with no additional computational cost. We show that the iterative solver can be stopped when the algebraic error is below the level of the estimated discretization error.

5.2 Perspectives

Several numerical tests have been performed for a benchmark symmetric Laplace problem, several steady and unsteady convection–diffusion, and Navier–Stokes problems solved with stabilized finite elements on highly stretched meshes. These test cases have been chosen for their straightforward implementation, to be used as future benchmarks on the subject. The results proved that, following the provided approach, the total number of iterations needed can be reduced up to five times, with no significant effect on the accuracy of the computed solution. This results in a considerable reduction of the computational time dedicated to the iterative solution of the problem.

Finally, a model problem of heat treatment in a 3D cavity was investigated to demonstrate the applicability of the developed approach to realistic full-scale

simulations. The numerical solver coupling Navier–Stokes and the heat transfer equation was run with adaptive stopping criterion based on two a posteriori error estimators. The computational was shown to be reduced favorably while retaining a good quality of the discrete solution with respect to the quantities of interest: this is promising for improving the performance of simulations of industrial applications in engineering.

The focus in this work is on a practical and general stopping criterion to reduce the CPU time required for the solution of coupled systems. However, the practically relevant industrial test cases to achieve higher treatment efficiency are not in the scope of this work. Therefore, in future works it is important to prove the usefulness of the proposed concept with a number of further real industrial cases.

Possible applications, where the features of the coupled solver are desirable, are for example simulations of quenching and cooling processes, or more challenging the simulation of multiphase f flows with phase changes. Indeed, such simulations are of great importance for the prediction and control of the ultimate microstructure of the workpieces but specially the reduction of both energy consumption and pollutant emissions.

Clearly, a number of other considerations have to be taken into account to complete this work and to achieve higher performance and robustness useful for an HPC-CFD framework. Here is the list of several important steps towards enhanced simulation tools for more realistic problems:

- The development of a more sophisticated error estimator taking into account boundary layers, wall function and high temperature interface gradients.
- A better determination of a stopping criterion for coupled problems. This particularly needed for turbulent flow coupled with two phase flow modeling and heat transfer.
- A deeper analysis and establishments of a clear relation between iterative algebraic solvers, resolution methods, error estimator and adaptive meshing, in particular for complex nonlinear settings.

RÉSUMÉ

L'objectif de ce travail est de proposer un critère d'arrêt pratique et général utilisant une approche a posteriori, qui s'appuie sur les estimations d'erreur disponibles à partir de la procédure d'adaptation du maillage. Ce critère d'arrêt doit être robuste et applicable aux différents types d'équations utilisées pour décrire la physique complexe impliquée dans un problème de transfert de chaleur conjugué. Le but final est de prouver qu'avec un tel critère d'arrêt il est possible de réduire drastiquement le temps CPU nécessaire à la résolution du système linéaire issu de la discrétisation Eléments Finis.

MOTS CLÉS

critère d'arrêt, estimations d'erreur, Eléments Finis.

ABSTRACT

The aim of this work is to propose a practical and general stopping criterion using an a posteriori approach, that relies on the error estimates available from the mesh adaptation procedure. This stopping criterion has to be robust and applicable to the different types of equations used to describe the complex physics involved in a conjugate heat transfer problem. The final goal is to prove that with such stopping criterion is possible to drastically reduce the CPU time required for the solution of the linear system that stems from the Finite Element discretization.

KEYWORDS

Stopping criteria, error estimator, finite elements.