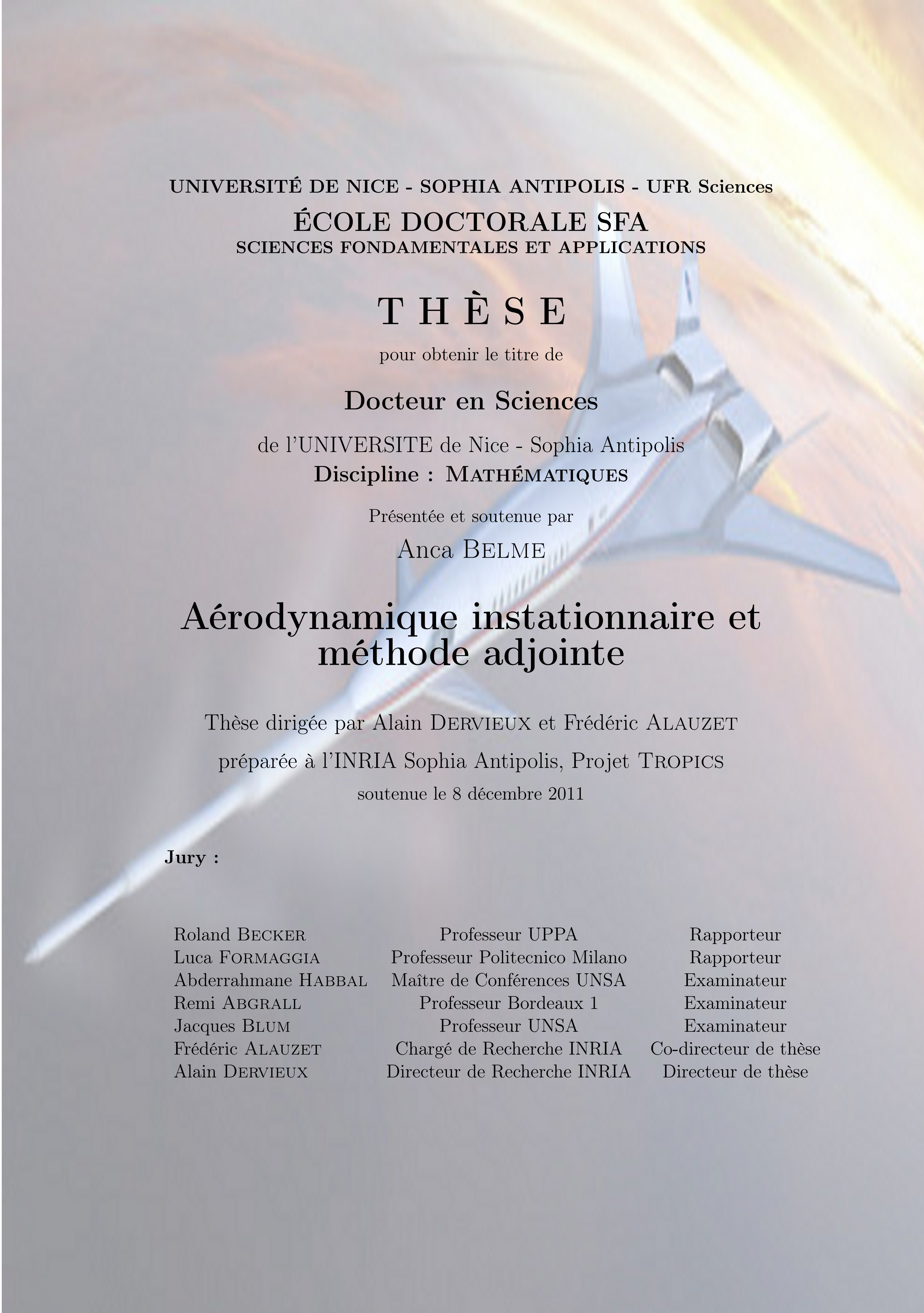


UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS - UFR Sciences

ÉCOLE DOCTORALE SFA

SCIENCES FONDAMENTALES ET APPLICATIONS

THÈSE

pour obtenir le titre de

Docteur en Sciences

de l'UNIVERSITE de Nice - Sophia Antipolis

Discipline : Mathématiques

Présentée et soutenue par

Anca Belme

Aérodynamique instationnaire et méthode adjointe

Thèse dirigée par Alain Dervieux et Frédéric Alauzet

préparée à l'INRIA Sophia Antipolis, Projet Tropics

soutenue le 8 décembre 2011

Jury :

Roland Becker	Professeur UPPA	Rapporteur
Luca Formaggia	Professeur Politecnico Milano	Rapporteur
Abderrahmane Habbal	Maître de Conférences UNSA	Examinateur
Remi Abgrall	Professeur Bordeaux 1	Examinateur
Jacques Blum	Professeur UNSA	Examinateur
Frédéric Alauzet	Chargé de Recherche INRIA	Co-directeur de thèse
Alain Dervieux	Directeur de Recherche INRIA	Directeur de thèse

Remerciements

Mes premiers remerciements vont vers mes deux directeurs de thèse, Alain Dervieux et Frédéric Alauzet, avec qui j'ai partagé trois années de recherche passionnantes.

Je remercie Alain Dervieux pour m'avoir permis de commencer la carrière dont j'ai toujours rêvé, pour son soutien et pour la transmission de son savoir et de son experience de chercheur. Je remercie aussi Frédéric Alauzet, qui représente pour moi un exemple d'enthousiasme et de rigueur scientifique, de m'avoir encadré tout au long de notre collaboration.

Je souhaite adresser d'autres remerciements à mes rapporteurs de thèse, Roland Becker et Luca Formaggia, qui ont pris de leur temps pour comprendre et analyser mes travaux, et aussi de faire partie du jury de ma thèse.

Je tiens à remercier également Remi Abgrall et Jacques Blum pour leur participation à mon jury. Je souhaite adresser un remerciement special pour Abderrahmane Habbal, pour ses conseils et sa collaboration au cours de ces années d'enseignement qui m'ont été trés utiles.

A mes collègues de l'équipe Tropics, mais également de l'équipe Gamma3, j'adresse un grand merci pour leur accueil et pour leur disponibilité.

Quant à mon collegue de bureau, Hubert Alcin, je le remercie d'avoir supporté mes humeurs parfois pétillantes.

Mon plus grand remerciement s'adresse à mon collègue de travail et mari, Alexandre Carabias, qui m'a soutenu et encouragé 24h/24h, et à ma famille et belle-famille, pour leur soutien et leurs encouragements sans faille.

Mes chers amis, je vous remercie egalement: Carmen, Geraldine, Afaf, Estelle, Audrey .... tout simplement pour avoir été là quand j'ai eu besoin de vous, merci !

"À la mémoire de mon père "

Contents

Chapter 1
Introduction

Contents

1.1 Introduction

The continuous fascination of scientists towards Fluid Dynamics is due to the fact that many flows that we can observe every day with our unaided senses pose deep scientific problems that have not yet been solved. Those unaided observations have led artists and scientifics wonder at the beauty, majesty and waywardness of flows over centuries.

The surface of any water puddle may be covered by a complex system of wrinkles (Figure 1.1 (left image)), who rudly represents the surrounding acoustic waves world. Other wrinkles, much more violent but still inconspicuous, enclose an engine travelling at a supersonic speed (Figure 1.1 (right image)) or these might be contained in shock waves produced by explosions. We notice especially the scale difference between the wave and the flow containing the wave.

During centuries the physical evolution of continuous environment has been modeled by equations (namely partial differential equations) leaving hope for a moment that their behaviour is fairly well approached by quite simple functions. The first resolution of partial differential equations used only a few degrees of freedom. However, we have to remark that the Fluid Dynamics engineer is now quite skeptic regarding such kind of simplifications.

This is especially true for a second family of multi-scale phenomenon, the turbulence. Turbulence was already recognized as a distinct fluid behaviour by at least 500 years ago. Figure 1.2 is an interpretation of one found in a sketch book of Da Vinci, along with a remarkably modern description:

> "... the smallest eddies are almost numberless, and large things are rotated only by large eddies and not by small ones, and small things are turned by small eddies and large."

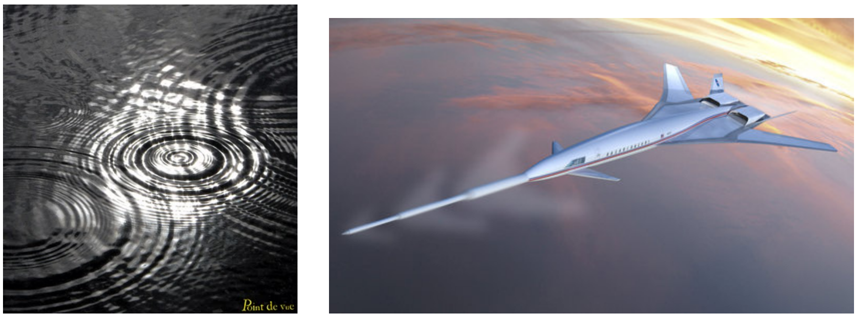


Figure 1.1: Sketch: Puddle of water which ripples (left image) and NASA supersonic futuriste airplane (right image).

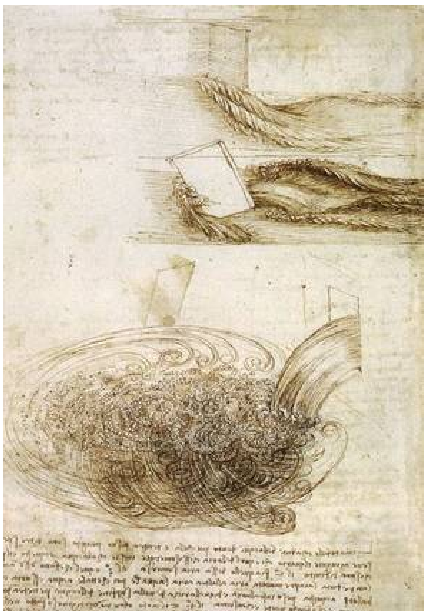


Figure 1.2: Da Vinci sketch of turbulent flow

It is a fact that most fluid flows are turbulent by nature, from the interior of biological cells, to circulatory and respiratory systems of living creatures, to countless technological devices and household appliances of modern society, to geophysical and astrophysical phenomena, including planetary interiors, oceans and atmospheres and stellar physics, and finally to galactic and even supergalactic scales (it has recently been proposed that turbulence during the very earliest times following the Big Bang is responsible for the present shape of the universe). Despite all this, the "problem of turbulence" remains to this days one of the last unsolved problem of classical mathematical physics.

The problem of turbulence has been studied by many of the greatest physicists and engineers of the 19th and 20th Centuries, and yet we do not understand in complete detail how or why turbulence occurs. Thus, study of turbulence is motivated both by its inherent intellectual challenge and by the practical utility of a thorough understanding of its nature.

The equations that are considered to embody the physics of turbulent flows, are the Navier-Stokes equations, introduced in the early of mid 19th Century and named by Navier and Stokes. As it will be shown later in this thesis, these equations are nonlinear and difficult to solve except in very particular cases. Analytical solutions that respects physics are generally not found, that is why we solve Navier-Stokes equations numerically.

The approximation of turbulent compressible flows:

State of the art: The simulation of unsteady turbulent flow demands a high quality for the numerical approximation. Most of the important academic calculations in the past were performed with higher order accurate approximation, generally on Cartesian meshes. These approximations cannot be applied on complex industrial geometries and as a result, the engineers use commercial software like Fluent which is based on unstructured second-order approximations. These approximations are robust and reliable but cannot produce efficient solution of Large Eddy Simulation models, which seem to require higher-order accuracy.

Our approach : The team in which I have been working in turbulence modeling has devoted several years to the development of a new generation of second-order approximations on unstructured meshes. The main characteristic of this generation is the use of a particular numerical dissipation, the upwind sixth-order viscosity (V6), which impacts the accuracy in a much smaller extend than usual fourth-order viscosities involved into second-order and third-order upwind approximations. The resulting scheme also enjoys a fifth-order superconvergence on Cartesian regions of the mesh. However, on very irregular meshes, the scheme can show an important deterioration in accuracy.

My contribution: In collaboration with Hilde Ouvrard, I have investigated the design of a version of the scheme advancing in time with a finite-element consistent mass matrix. The problem is the stabilisation of this. We proposed a sixth-order numerical viscosity and analysed the properties of accuracy and stability of the new scheme. A comparison for a LES flow with the V6 initial scheme has been performed.

The modelisation of high Reynolds turbulent flows :

State of the art: The modelisation and numerical prediction of high Reynolds unsteady turbulent flows is the challenge of these decades. Statistical turbulence modeling is a topic in a consolidation phasis. Large Eddy Simulation (LES) seems blocked by the necessary accounting of a very large part of turbulence energy in resolved scales, rendering LES too costly in industry, in particular for flows with boundary layers. A new generation of engineering models combine a statistical model (typically near boundaries) with a LES one. These hybrid formulations, like the Detached Eddy Simulation (DES) of Travin *et. al* [Travin 1999], allows unsteady high Reynolds predictions. However, the first versions of hybrid schemes involve simplified ingredients (one-equation statistical component, Smagorinsky-like subgrid scale model). The way they work needs more understanding.

Our approach : The team in which I have been working in turbulence modeling is studying the combination of sophisticated ingredients in an hybrid formulation. This involve a low Reynolds two-equation model, but also a particular wall law, for the statistical component. For the LES component, a VMS-LES method is used.

My contribution: I have studied the behavior of a hybrid model involving a dynamic monitoring of the VMS-LES term. The study was concentrating on the flow past a cylinder for very high Reynolds numbers, between 0.5 Million and 3 Million.

The estimation of numerical errors:

State of the art: In numerical analysis, managing numerical errors is of paramount importance. Indeed, all computational results obtained using mathematical models of an event will involve numerical errors. The knowledge of such errors can provide a means for assessing the reliability of the computation, as well as a basis for adaptive control of the numerical process. The first use of error estimates for adaptive meshing strategies in significant engineering problem was given in the work of Guerra [Guerra 1977] in 1977. A posteriori estimates deals, in a majority of cases, with global estimates in finite element approximations for elliptic problems. In the 1990's techniques for computing local estimates and estimates of errors in "quantities of interest" began to appear, where, such quantities of interest manifest themselves as functionals. These estimates provided the basis of "goal-oriented adaptivity" wherein adaptive meshing procedures are devised to control error in quantities of interest. A general setting of such technique was proposed for example in [Becker 1996a], [T. Strouboulis 2000]. A general account of such duality problems, with applications to both linear and non-linear problems was contributed by [Becker 2001].

Our approach : Our team focuses on error estimators for mesh adaptation problems and error correction methods. Loseille [Loseille 2008] has shown in his thesis how an *a priori* estimate based on interpolation errors in L^2 space can be developed for generating anisotropic adapted meshes for steady Euler problems and reaching second-order convergence rate for problems with shocks.

My contribution: In this thesis I focused on error estimates for unsteady problems. I have contributed to both *a posteriori* and *a priori* error estimators for unsteady inviscid problems and viscous unsteady problems. For the first one, we have been interested on linearized methods for reducing dissipation errors. Regarding the *a priori* errors, a new estimator is proposed with application to viscous flows. These *a priori* estimators have been employed for goal-oriented anisotropic mesh adaptation problems, for both Euler and laminar Navier-Stokes flows.

Applications to mesh adaptation:

State of the art: Mesh adaptation methods are closed to reaching a remarkable maturity and efficiency. If the isotropic adapted meshes were often prefered in the past, it has become clearer and clearer that the use of anisotropic meshes must be recommended for realistic aerodynamics, since the resolution near the boundary is the most costly and sensitive to discretization, and anisotropic meshes improve this issue.

Our approach : Our team, in strong cooperation with Gamma Project from INRIA Rocquencourt, studies methods in which anisotropic meshes are described by metrics defined in a continuous framework. In this context, the error model is bounded by interpolation errors defined by Hessians of the interpolated variable, weighted by some weights. Working in this continous framework permits an analytic resolution of the optimisation problem.

My contribution: My joint work with Gamma team referred to application of their anisotropic mesh generators to functionals of observation. More exactly, we have developed a method to derive an optimal mesh to observe/improve a given output functional in an unsteady context. The weights of the interpolation error are adjoint states in this case. A new global fixed-point algorithm is proposed here in order to converge the couple mesh/solution. We have applied this algorithm for blast wave problems and acoustics, for both 2D and 3D cases.

1.2 Organisation and Content of thesis

The present thesis is built around two main topics:

- First, the understanding and the improvement of turbulent flow simulation and modelisation through a parallel platform dealing with turbulent Navier-Stokes system coupled with turbulence models.
- And second, managing approximation errors through an adjoint-based formulation, either to improve a target error or through a goal-oriented anisotropic mesh adaptation process. The final goal of this second part is the construction

of a solid error estimation environment able to deal with massively separated turbulent flows.

The first two chapters of the present thesis refers mainly to our first purpose, the turbulent flow simulation and modelisation, although in the remaining chapters the turbulence can be evoked. The remaining parts of this thesis concentrates on numerical error reduction.

- **Chapter 2** starts with a short review of the Numerical Schemes employed in this thesis. A new mass matrix formulation of this scheme is introduced and tested for a 2D Gaussian translation problem.

- **Chapter 3** refers to high Reynolds number turbulent flows simulation. We present an overview of some well-known turbulence models, focusing on hybrid ones. Results for the simulation of the flow around a circular cylinder at subcritical Reynolds number for the new mass matrix scheme are presented. Then we study a new hybrid model and apply it to high Reynolds flow computation.

- **Chapter 4** is an introductory chapter to the second subject of this thesis, namely output-based mesh adaptation. We introduce the adjoint state, for both continuous and discrete formulations. An example of adjoint computation is also described before ending this chapter.

- **Chapter 5** is dedicated to error correction methods, focusing on *a posteriori* estimates. Error correction methods based on linearization and adjoints are discussed here, and regarding the linearised methods, a numerical example is provided for reducing the dissipation error.

- **Chapter 6** contains the theoretical developement of an *a priori* error estimate based on a finite-element formulation. It is shown how approximation errors can be bounded by interpolation errors through this *a priori* analysis.

- **Chapter 7** presents the continuous framework for the parametrization by metrics of meshes. The concept of continuous metric and associated mesh are reviewed here.

- **Chapter 8** is dedicated to the description and resolution of the mesh adaptation problem for unsteady Euler flows. We concentrate on output-based formulation, showing how the error model is described in a continuous metric-based framework. A global fixed-point algorithm is presented here to converge the unsteady problem mesh/solution. This chapter ends with applications to blast waves and acoustics problems.

- **Chapter 9** is a preliminary chapter devoted to output-based mesh adaptation problem associated with Navier-Stokes viscous flows. The error estimate developed in Chapter 6 is used to solve the minimization problem. The numerics are identical to the ones in Chapter 8. The drag computation for the flow around a NACA0012 airfoil is optimised here, through this goal-oriented mesh adaptation model.

1.3 Scientific communications

• • Publications • •

Articles accepted in International Journals :

AD-based perturbation methods for uncertainties and errors, M. Martinelli, A. Dervieux, L. Hascoet, V. Pascal, A. Belme, *International Journal Engineering Systems Modelling and Simulation*, 2010, vol.2, No. 1/2, pp. 65-74.

Articles submitted in International Journals :

Time Accurate Anisotropic Goal-Oriented Mesh Adaptation for Unsteady Flows, A.Belme, A. Dervieux, F. Alauzet, sent to *Journal of Computational Physics.*

Research Report:

Combining a Mass Matrix formulation and a high order dissipation for the discretisation of turbulent flows, A.Belme, H.Ouvrard, Research Report INRIA, http://hal.inria.fr (available online), 2008

• • Communications • •

Conferences, Workshop, Colloquium:

1. **Anisotropic goal-oriented mesh adaptation for time dependent problems**, F. Alauzet, A. Belme and A. Dervieux, 20th IMR Conference, Springer, Paris 2011

2. **Application of anisotropic goal-oriented unsteady mesh adaptation to Aerodynamics and Aeroacoustics**, A. Belme, A. Dervieux, F. Alauzet, Adaptive Modelling and Simulation ADMOS 2011 Conference, Paris 2011

3. **A priori anisotropic goal-oriented estimates for mesh adaptation in compressible CFD**, A. Belme, F. Alauzet. A. Dervieux, A. Loseille, 16th International Conference on Finite Elements in Flow Problems, march 2011

4. **Application of hybrid and VMS-LES turbulent models to aerodynamic simulations**, A.Belme, A.Dervieux, B.Koobus, S.Wornom, M.V.Salvetti, ICAS Congress , Nice, September 2010

5. **Mesh-adaptive computation of linear and non-linear acoustics**, A. Belme, A. Dervieux, F. Alauzet, Trilateral Seminar "Computational experiment in aeroacoustics", Svetlogorsk, Russia, September 2010

6. **Goal-Oriented Anisotropic Mesh Adaptation for Unsteady Flows**, A. Belme, A. Dervieux and F. Alauzet, ECCOMAS CFD Conference, Lisbon, june 2010

7. **Correcteurs des erreurs numérique par linéarisations**, A.Belme, Proceeding SMAI, La Colle sur Loup, France (poster), 2009

Chapter 2

Numerical aspects

Contents

2.1 Introduction

This chapter is devoted to the Numerical Schemes which will be used in the different chapters of this thesis, and in particular for the hybrid simulation of large eddies in turbulent flows, for acoustics and for shock motions. The common feature of these models is the compressibility of the fluid.

A stable scheme need to be derived for these convected dominated flows, with the ability of shock capturing. Beside this, the convection part needs in most cases to involve a numerical dissipation which is as small as possible. Several types of schemes with six-order dissipation are considered. The low level of dissipation makes these scheme better adapted to Large Eddy Simulation (LES) than usual third-order upwind accurate schemes.

One of the contribution of this thesis is the building of a new scheme involving a consistent mass matrix formulation stabilised with such a sixth-order dissipation. All our schemes are based on a central finite-element type approximation allowing an error analysis developed in Chapter 6.

This chapter is organized in three sections. We start with Section 2.2 where we focus on the description of main numerical aspects for spatial resolution of a simpler model, the Euler model. In this section one will be guided through the spatial representation and flux integration of the Euler system of equations. Next, in Section 2.3 we shall focus on higher order representation of the spatial resolution. Section 2.4 introduces and analyses a new mass matrix formulation for improving the accuracy on very irregular meshes. A first numerical application of this scheme is presented here. This section represents also my first, introductory contribution to the numerical framework of our team. The last part of this chapter, that is Section 2.5, is dedicated to the implementation of these upper mentioned schemes to Navier-Stokes system of equations.

2.2 Numerical Method for 3D Euler equations

2.2.1 Mathematical model for Euler equations

For a perfect, inviscid gas, the unsteady Euler equations for mass, momentum and energy writes in a conservative formulation :

$$\Psi(W) = \frac{\partial W}{\partial t} + \nabla \cdot \mathcal{F}(W) = 0 \quad \text{in} \quad \Omega, \tag{2.1}$$

where $\Omega \subset \mathbb{R}^3$ is the computational domain, $W = {}^t(\rho, \rho u, \rho v, \rho w, \rho E)$ is the vector of conservative variables, $\frac{\partial W}{\partial t}$ is the time derivative (an usual notation could be also:

W_t), and $\mathcal{F}$ is the convection operator $\mathcal{F}(W) = (\mathcal{F}_1(W), \mathcal{F}_2(W), \mathcal{F}_3(W))$ with:

$$\mathcal{F}_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{pmatrix}, \mathcal{F}_2(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \end{pmatrix}, \mathcal{F}_3(W) = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho E + p)w \end{pmatrix}, \quad (2.2)$$

where ρ, p and E holds respectively for the density, the thermodynamical pressure and the total energy per unit mass. Symbols u, v and w stands for the Cartesian components of the velocity vector $\mathbf{u} = (u, v, w)$. For a calorically perfect gas, we have

$$p = (\gamma - 1)\big(\rho E - \frac{1}{2}\rho|\mathbf{u}|^2\big),$$

where γ is constant.
We impose corresponding boundary conditions, for instance slip conditions on the solid boundary denoted here Γ:

$$\mathbf{u} \cdot \mathbf{n} = 0$$

where $\mathbf{n}$ is the outward normal to Γ, and we consider the flow to be uniform on the far-field boundary Γ_∞. At initial time we impose also: $W(\mathbf{x}, 0) = W_0(\mathbf{x}),\ \forall \mathbf{x} \in \Omega$.

System (2.1) has the property of homogeneity, which means that functions $\mathcal{F}_i$, for $i = 1, 2, 3$ verifies the following property : $\mathcal{F}_i(\lambda W) = \lambda \mathcal{F}_i(W)$. This can be translated into the following relations: $\mathcal{F}_i(W) = \mathcal{F}_i^{'}(W)W$.

Definition 2.2.1
A system of PDEs like 2.1 is called hyperbolic for the direction t if the associated characteristic matrix denoted H (in this case $H = \eta_1\mathcal{F}_1^{'} + \eta_2\mathcal{F}_2^{'} + \eta_3\mathcal{F}_3^{'}$) has real eigenvalues and if the corresponding eigenvectors are linearly independent for all η_1, η_2, η_3.

Proposition 2.2.1
Euler System (2.1) (with boundary and initial conditions) is an **hyperbolic** *system for the direction t on the computational domain Ω.*

We recall that the Euler System (2.1) generally does not admit a global regular solution, even if the initial state $W_0(x)$ is regular. Discontinuities, for instance a shock wave, can develop at some time. In order to consider discontinuous solutions, we introduce a weak solution.

A weak formulation of System (2.1) writes:

$$\begin{aligned} \forall \phi \in C^\infty(\bar{\Omega} \times \bar{R}^+), \quad (\Psi(W)\,,\,\phi) \;=\; & \int_0^T \int_\Omega [-W\ \phi_t - \mathcal{F}(W)\nabla\phi]\ \mathrm{d}\Omega \mathrm{d}t \\ & + \int_\Omega [\phi(T)W(T) - \phi(0)W(0)]\ \mathrm{d}\Omega \\ & + \int_0^T \int_\Gamma \phi(\hat{\mathcal{F}}(W) + \mathcal{F}(W)) \cdot \mathbf{n}\ \mathrm{d}\Gamma \mathrm{d}t = 0, \end{aligned} \quad (2.3)$$

where $\hat{\mathcal{F}}(W) + \mathcal{F}(W)$ contains the boundary conditions and boundary flux.

Definition 2.2.2
(**Rankine-Hugoniot relation**) *Let W be a discontinuous solution of System (2.1) along a regular hypersurface, defined for all couple (x,t) and with normal vector $(-\sigma, \mathbf{n})$. We denote W_1 and W_2 two states on respectively the left and the right size of the discontinuity. The relation of Rankine-Hugoniot writes:*

$$\|\mathcal{F}(W_1) - \mathcal{F}(W_2)\| \cdot \mathbf{n} = \sigma\,[W_1 - W_2]$$

where $\mathbf{n}$ and σ are respectively direction and velocity of propagation of the travelling discontinuity.

Unicity of the weak solution for System (2.3) does not holds, but not all of these solutions have a physical interpretation. An example of accepted solutions are defined by the limit when $\varepsilon \longrightarrow 0$ of viscous solutions of the following system:

$$W_t + \nabla \cdot \mathcal{F}(W) = \varepsilon \Delta W,\ \varepsilon > 0.$$

In a simplified mono-dimensional context it means that this weak solution need to satisfy the entropy condition that we recall hereafter:

Definition 2.2.3
A convex function η is called an **entropy** *associated to Euler System (2.1) if functions q_1, q_2, q_3 exists (also called entropy flux) and such that:*

$$\frac{d\eta(W)}{dW}\frac{dq_k(W)}{dW} = \frac{d\mathcal{F}_k(W)}{dW} \quad for \quad k = 1,2,3.$$

A weak solution W of System (2.1) is called **entropy solution** *if for any entropy η of flux q_1, q_2, q_3 associated to (2.1), it verifies the following inequality:*

$$\frac{\partial \eta(W)}{\partial t} + \sum_{k=1}^{3} \frac{\partial q_k(W)}{\partial x_k} \leq 0.$$

In practice, our numerical schemes will involve a numerical dissipation which will be sufficient for ensuring to obtain the unique approximations of the entropic solution.

2.2.2 A particular P1 Finite-Element Galerkin formulation

Let Ω_h be a polygonal domain close to Ω. We assume that Ω_h can be discretized by a tetrahedrization $\mathcal{T}_h$ admissible for Finite-Elements *i.e.*, the intersection of two different tetrahedra is either empty, or a vertex, or an edge, or a face. We shall call Ω_h the discrete computational domain, and Γ_h the associated discrete boundary.

The test functions are taken into the approximation space V_h made of continuous piecewise linear functions included in $V = [H^1(\Omega)]^5$:

$$V_h = \left\{ \phi_h \,\middle|\, \phi_h \text{ is continuous and } \phi_{h|T} \text{ is linear } \forall T \in \mathcal{T}_h \right\} .$$

In order to avoid the management of interpolation operators applicable in the whole H^1 space, we shall work inside the following spaces:

$$\bar{V} = [H^2(\Omega)]^5 \;\text{ and }\; \bar{V}_h = \bar{V} \cup V_h \,.$$

This is useful for using Π_h, the standard P^1 interpolation operator:

$$\begin{aligned} \Pi_h : \bar{V}_h &\longrightarrow V_h \\ \phi &\longmapsto \Pi_h \phi \quad \text{ with } \;\; \Pi_h\phi(i) = \phi(i) \;\; \forall i \text{ vertex of } \mathcal{T}_h \,. \end{aligned}$$

Then for Problem (2.3), a discrete unsteady Euler formulation writes:

$$\forall \phi_h \in V_h, \;\; \int_{\Omega_h} \left(\phi_h W_{h,t} + \phi_h \nabla . \mathcal{F}_h(W_h) \right) \, d\Omega_h - \int_{\Gamma_h} \phi_h \hat{\mathcal{F}}_h(W_h) . \mathbf{n} \; d\Gamma_h \; = \; 0 \,, \tag{2.4}$$

where $\mathcal{F}_h$ is by definition the P^1 interpolate of $\mathcal{F}$, in the sense that:

$$\mathcal{F}_h(W) = \Pi_h \mathcal{F}(W) \;\text{ and }\; \mathcal{F}_h(W_h) = \Pi_h \mathcal{F}(W_h) \,, \tag{2.5}$$

and, as operator $\mathcal{F}_h$ applies to values of W at mesh vertices, we have:

$$\mathcal{F}_h(W) = \mathcal{F}_h(\Pi_h W) = \Pi_h \mathcal{F}(\Pi_h W) \,. \tag{2.6}$$

We use the same relations for defining $\hat{\mathcal{F}}_h(W)$:

$$\hat{\mathcal{F}}_h(W) = \Pi_h \hat{\mathcal{F}}(\Pi_h W) \;\text{ and }\; \hat{\mathcal{F}}_h(W_h) = \Pi_h \hat{\mathcal{F}}(W_h) \,. \tag{2.7}$$

Practically, this definition of $\mathcal{F}_h$ means that in $\mathcal{F}_h$, nodal fluxes values of W are first evaluated at the mesh vertices. Consequently, discrete fluxes are derived from these nodal values by P^1 extrapolation inside every element. In contrast to the standard Galerkin approach, this definition emphasizes that the fluxes are projected in V_h.

2.2.3 Mixed-Element-Volume basic equivalence

The discrete Formulation (2.4) can be transformed into a vertex-centered finite-volume scheme applied to tetrahedral unstructured meshes. This is donne using the mixed element-volume approach as in [Stoufflet 1996, Selmin 1998]. This assumes a particular partition in control cells C_i of the discretized domain Ω_h:

$$\Omega_h = \bigcup_{i=1}^{n_c} C_i \;, \tag{2.8}$$

each control cell being associated with a vertex i of the mesh.
The *dual finite-volume cell* is built by the rule of medians. In 2D, the median cell is limited by segments of medians between centroids and mid-edge (Figure 2.1). In 3D, each tetrahedron T of the mesh is split into four hexahedra constructed around each of its four vertices. For a vertex i, the polyhedron $C_i \cap T$ is defined by the following points (Figure 2.2):

(i) the three middle points of the edges of T issued from i,

(ii) the three gravity centers of the faces of T containing i,

(iii) the center of gravity of the tetrahedron and

(iv) the vertex i.

The cell C_i of vertex i is the collection of all polyhedra linked to i. The common boundary $\partial C_{ij} = \partial C_i \cap \partial C_j$ between two neighboring cell C_i and C_j is decomposed in several triangular interface facets. An illustration of this construction is shown in Figure 2.3 for the 3D case.

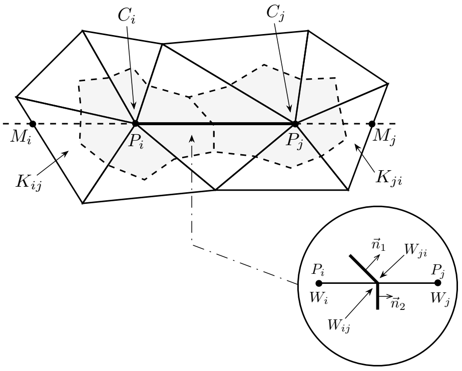


Figure 2.1: Illustration of finite-volume cells construction in two dimensions with two neighboring cells, C_i and C_j around i (P_i on the figure) and j (P_j on the figure) respectively, and of the upwind triangles K_{ij} and K_{ji} associated with the edge ij.

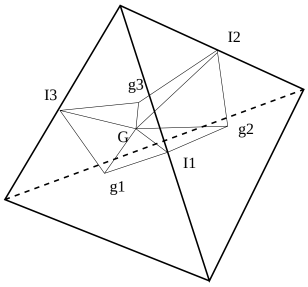


Figure 2.2: The planes which delimit finite-volume cells inside a tetrahedron (3D case).

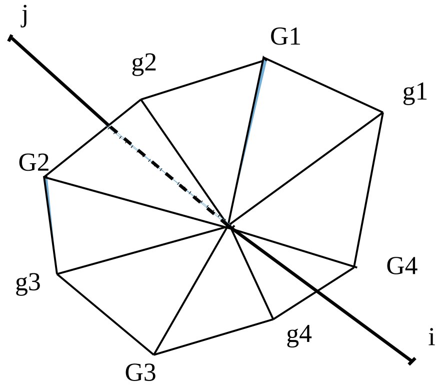


Figure 2.3: Illustration for 3D case of finite-volume cell interface ∂C_{ij} between two neighboring cells C_i and C_j.

The corresponding test functions are the piecewise constant characteristic functions of cells:

$$\chi^i(x) = \begin{cases} 1 & \text{if } x \in C_i, \\ 0 & \text{otherwise.} \end{cases}$$

Then the weak form of Euler equations (2.1) is integrated in this new formulation by writing, for each vertex i, *i.e.* for each cell C_i:

$$\int_{C_i} W_t \; dC \quad + \quad \int_{\partial C_i} \mathcal{F}(W)\mathbf{n}_i \; d\sigma = 0 \tag{2.9}$$

The finite-volume fluxes between cells around vertices i and j are integrated through the common boundary ∂C_{ij} with a value of $\mathcal{F}_h$ equal to the half-sum of $\mathcal{F}_h(W_i)$ and $\mathcal{F}_h(W_j)$:

$$\Phi_{ij}^{MEV} = \frac{\mathcal{F}_h(W_i) + \mathcal{F}_h(W_j)}{2} \, . \, \nu_{ij} \, , \tag{2.10}$$

where ν_{ij} denotes the integral of the normal $\mathbf{n}_i$ to common boundary between cells C_i and C_j,

$$\nu_{ij} = \int_{\partial C_{ij}} \mathbf{n}_i \; d\sigma$$

and $W_i = W(i)$. The finite-volume formulation for an internal vertex i writes as the sum of all the fluxes evaluated from the vertices j belonging to $V(i)$ where $V(i)$ is the set of all neighboring vertices of i. Taking into account the boundary fluxes, the discrete Scheme (2.4) then writes:

$$\sum_{j \in V(i)} \Phi_{ij}^{MEV} \; - \; \int_{\Gamma \cap \partial C_i} \bar{\mathcal{F}}_h(W_h).\mathbf{n} \; d\Gamma \; = \; 0 \, . \tag{2.11}$$

We obtain a vertex-centered finite-volume approximation which is P^1-exact with respect to the flux function $\mathcal{F}_h$. This scheme enjoys most of the accuracy properties of the Galerkin method [Mer 1998], such as the second-order accuracy on any mesh for diffusion-convection models. However, it lacks stability and cannot be applied to purely hyperbolic models such as the Euler equations. Median-based cells are the exact counterbalance of the P^1 finite element formulation. They are well adapted to non-stretched unstructured meshes. In the case of highly stretched meshes, upwind finite volume methods show a truncation error which grows with aspect ratio. In some other case, cells with shapes closer to rectangles may allow a higher accuracy. Barth [Barth 1994] introduced the circumcenter cells as an issue for the encountered problems. We can also apply a barycenter for mixing the two cells types, circumcenter ones for very stretched parts and medians for less streched ones (not used in this thesis).

2.2.4 Flux integration

Once the cells are defined, the spatial divergence $\nabla \cdot \mathcal{F}$ is transformed via the Stokes formula into integrals of normal fluxes $\mathcal{F}.\mathbf{n}$ at cell boundaries. In the proposed family of schemes, the accuracy of the integral geometrical quadrature is not as crucial. We define a very simple option, the *edge-based integration.* At the contrary, flux integration sets the important problem of scheme stabilisation. The variables are assumed to be constant by cell, and therefore, they are discontinuous from a cell to its neighbor. Upwind integration of the discontinuous fluxes will rely on the Godunov method based on the two different values of the unknown at each side of the discontinuity.

2.2.4.1 Central differencing

Let us write a vertex-centered central differenced finite-volume scheme for the Euler equations applied to an unstructured mesh as follows:

$$\begin{aligned} \mathbf{\Psi}_h(\gamma, W)_j &= 0, \quad \text{with} \\ \mathbf{\Psi}_h(\gamma, W)_j &= \sum_{k \in V(j)} \Phi^{central}(W_j, W_k, \nu_{jk}) \; + \; \mathbf{B}_h(\gamma, W)_j \end{aligned} \tag{2.12}$$

where $V(j)$ is the set of vertices that are neighbors of j, ν_{jk} is the integral on interface between j and k of the normal vector. Symbol $\mathbf{B}_h(\gamma, W)_j$ holds for boundary fluxes. The centered integration for elementary flux Φ is written as follows:

$$\Phi^{central}(W_j, W_k, \nu_{jk}) \;=\; 0.5(\mathcal{F}_j + \mathcal{F}_k).\nu_{jk} \tag{2.13}$$

where $\mathcal{F}_j = \mathcal{F}(W_j)$ are the Euler fluxes computed at W_j. This is equivalent to introducing the following discrete space operator ∇_h^*:

$$\nabla_h^*(f)_j \;=\; \frac{1}{|C_j|} \sum_{k \in V(j)} (f_j + f_k)/2 \;\; \nu_{jk} \tag{2.14}$$

where $|C_j|$ is the measure of cell C_j.

2.2.4.2 Godunov differencing

Central finite-volume (FV) methods are numerically instable. In the general case they do not produce a solution, and when they produce a solution, it does not always satisfy the entropy condition. Starting from the availability of analytical solutions of Riemannian Problem, Godunov has proposed to use them in combination with the discontinous representation of the solution.

Godunov-type methods rely on discontinuous representations of the unknown and computation of fluxes at discontinuities in function of both "left" and "right" values by applying an approximate or an exact Riemann solver. This process introduces numerical viscosity terms that are very useful for stabilizing flows involving shocks.
We write a vertex-centered first-order Godunov scheme for the Euler equations applied to an unstructured mesh as follows:

$$\mathbf{\Psi}_h(\gamma, W)_j \;=\; \sum_{k\in V(j)} \Phi(W_j, W_k, \nu_{jk}) \;+\; \mathbf{B}_h(\gamma, W)_j. \tag{2.15}$$

The upwinding in elementary flux Φ is the Roe flux splitting. In the case of the standard Roe splitting, we have:

$$\Phi(W_j, W_k, \nu_{jk}) \;=\; 0.5(\mathcal{F}_j + \mathcal{F}_k).\nu_{jk} \quad + \quad 0.5|\mathcal{A}|(W_j - W_k) \tag{2.16}$$

where $|\mathcal{A}|$ is the absolute value of the Jacobian flux along ν_{jk}:

$$\begin{gathered}\mathcal{A} = (\tfrac{\partial\mathcal{F}}{\partial W})_1(\nu_{jk})_1 \;+\; (\tfrac{\partial\mathcal{F}}{\partial W})_2(\nu_{jk})_2 \\ \mathcal{A} = T\Lambda T^{-1}\ ,\quad \Lambda \quad diagonal, \\ |\mathcal{A}| = T|\Lambda|T^{-1}\ .\end{gathered} \tag{2.17}$$

These matrices are computed at an intermediate value $\overline{W}_{jk}$ of W: of W_j and W_k, in short:

$$\overline{W}_{jk} \;=\; \left(\rho_j^{\frac{1}{2}} W_j + \rho_k^{\frac{1}{2}} W_k\right) / (\rho_j^{\frac{1}{2}} \rho_k^{\frac{1}{2}})$$

which enjoys the following property:

$$\mathcal{F}(W_j) - \mathcal{F}(W_k) \;=\; \mathcal{A}(\overline{W}_{jk})(W_j - W_k)\ .$$

In (high enough) supersonic case, $\mathcal{A}(\overline{W}_{jk}) = |\mathcal{A}(\overline{W}_{jk})|$ or $\mathcal{A}(\overline{W}_{jk}) = -|\mathcal{A}(\overline{W}_{jk})|$ and Roe's splitting is fully upwind. By the hyperbolicity assumption, matrix A can be diagonalised. The absolute value $|\mathcal{A}|$ writes:

$$|\mathcal{A}(\overline{W}_{jk})| \;=\; T^{-1}Diag(|\lambda_1|, |\lambda_2|, |\lambda_3|, |\lambda_4|)T \;=\; sign(\mathcal{A})\mathcal{A}\ ,$$

where

$$sign(\mathcal{A}) \;=\; T^{-1}Diag(sign(\lambda_1), sign(\lambda_2), sign(\lambda_3))T. \tag{2.18}$$

Thus this averaging also permits the following equivalent formulation:

$$(\mathcal{F}(W_j) + \mathcal{F}(W_k))/2 \;=\; sign(\mathcal{A}(\overline{W}_{jk}))(\mathcal{F}(W_R) - \mathcal{F}(W_L))/2\ . \tag{2.19}$$

where W_R and W_l are the values of the unknown ar the right and respectively left of the discontinuity.
These schemes are spatially first-order accurate.

2.3 Towards higher spatial order

First-order upwind schemes of Godunov type enjoy a lot of interesting properties and in particular monotonicity or, more or less equivalently, positivity. They can be extended to second order accuracy by applying the MUSCL method. We give in the next subsection a description of how this can be done. Unfortunately, even for the second-order version, the amount of dissipation which is introduced seems larger than that needed in many applications.

In the second subsection, we recall a method inspired by Direct Simulation techniques in which non-dissipative high-order approximations are stabilised in good accuracy conditions thanks to filters which rely on very-high even order derivatives. When done in ad-hoc manner, this leads to superconvergent advection schemes which are more accurate for a subclass of applications, typically, for linear hyperbolic system solved on Cartesian meshes.

2.3.1 The second order MUSCL method

The Godunov method builds fluxes between cells in which unknown variables are considered as constant. This results in a first-order accurate scheme, not enough accurate for most applications. Van Leer has proposed ([Leer 1979, Leer 1977]) to reconstruct a linear interpolation of the variables inside each cell and then to introduce in the Riemann solver the boundary values of these interpolations. Further, the slopes used for linear reconstruction can be limited in order to represent the variable without introducing new extremas. The resulting MUSCL method produces positive second-order schemes. We describe now an extension of MUSCL to unstructured triangulations with dual cells. The MUSCL ideas also applies to reconstructions which are different on each interface between cells, or equivalently on each edge. Several slopes of a primitive variable $U = (\rho, u, v, p)$ are defined on the two vertices i and j of an edge ij as follows :

- First, the *centered gradient* $(\nabla U)^c_{ij}$ is defined as:

$$(\nabla U)^c_{ij} \,.\, \vec{ij} \;=\; U_j \;-\; U_i. \tag{2.20}$$

- We consider a couple of two triangles, one having i as a vertex, and the second having j as a vertex. With reference to Figure 2.4, we define ε_{ni}, ε_{mi}, ε_{jr} and

ε_{js} as the components of vector $\vec{ji}$ (resp. $\vec{ij}$) in the oblique system of axes $(\vec{in}, \vec{im})$ (resp. $(\vec{jr}, \vec{js})$):

$$\vec{ji} = \varepsilon_{ni}\, \vec{in} + \varepsilon_{mi}\, \vec{im}\,,$$

$$\vec{ij} = \varepsilon_{jr}\, \vec{jr} + \varepsilon_{js}\, \vec{js}\,.$$

- We shall say that T_{ij} and T_{ji} are upstream and downstream elements with respect to edge ij if the components $\varepsilon_{ni}, \varepsilon_{mi}, \varepsilon_{jr}, \varepsilon_{js}$ are all nonnegative:

$$T_{ij} \;\text{ upstream and } T_{ji} \;\text{ downstream } \;\Leftrightarrow Min(\varepsilon_{ni}, \varepsilon_{mi}, \varepsilon_{jr}, \varepsilon_{js}) \geq 0.$$

The *upwind gradient* $(\nabla U)^u_{ij}$ is computed as the usual finite-element gradient on T_{ij} and the *downwind gradient* $(\nabla U)^d_{ij}$ on T_{ji}. This writes :

$$(\nabla U)^u_{ij} \;=\; \nabla U\,|_{T_{ij}} \text{and} (\nabla U)^d_{ij} \;=\; \nabla U\,|_{T_{ji}} \tag{2.21}$$

where $\nabla U\,|_T \;=\; \sum_{k\in T} U_k \nabla\Phi_k|_T$ are the P1-Galerkin gradients on triangle T.

We now specify our method for computing the *extrapolation slopes* $(\nabla\, U)_{ij}$ and $(\nabla\, U)_{ji}$:

$$(\nabla U)_{ij}.\vec{ij} = \quad (1-\beta)(\nabla U)^c_{ij}.\vec{ij} + \beta(\nabla U)^u_{ij}.\vec{ij}\ . \tag{2.22}$$

The computation of U_{ji} is analogous:

$$(\nabla U)_{ji}.\vec{ij} = \quad (1-\beta)(\nabla U)^c_{ij}.\vec{ij} + \beta(\nabla U)^d_{ij}.\vec{ij}\ . \tag{2.23}$$

The coefficient β is an upwinding parameter that controls the combination of fully upwind and centered slopes and that is generally taken equal to 1/3, according to the error analysis below.

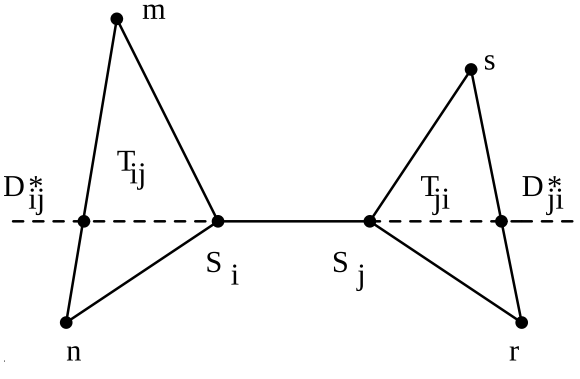


Figure 2.4: Localisation of the extra interpolation points D^*_{ij} and D^*_{ji} of nodal gradients.

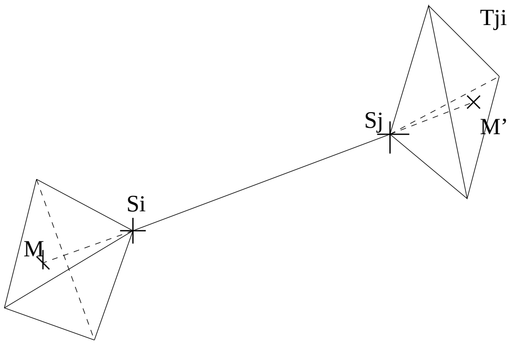


Figure 2.5: Downstream and Upstream tetrahedra are tetrahedra having resp. S_i and S_j as a vertex and such that line S_iS_j intersects their opposite face.

The scheme description is completed by replacing the first-order formulation (2.15) by the following flux balance:

$$\mathbf{\Psi}_h(\gamma, U)_j \;=\; \sum_{k \in V(j)} \Phi(U_{jk}, U_{kj}, \nu_{jk}) \;+\; \mathbf{B}_h(\gamma, U)_j. \tag{2.24}$$

with

$$U_{jk} = U_j + \frac{1}{2}(\nabla U)_{ij}.\vec{ij} \quad , \quad U_{kj} = U_k + \frac{1}{2}(\nabla U)_{ji}.\vec{ji}\ .$$

This approximation is spatially second-order accurate for regular meshes. It is known that in the unsteady case and for very irregular meshes, the scheme accuracy degrades to less than 2, in a similar manner to finite-element with mass lumping. Even for regular meshes, for a nonlinear flux function the accuracy of MUSCL schemes is limited to second order, as remarked by Wu and Wang [Wu 1995]. Further, the method combines finite differences in the local reconstruction and finite volume for fluxes. As a consequence, improving the reconstruction to higher order interpolation does not carry a higher accuracy. We now examine how to change the reconstruction in order to improve the scheme.

2.3.2 Low dissipation advection schemes : 1D

2.3.2.1 Spatial 1D MUSCL formulation

Let us first consider the one-dimensional scalar conservation law

$$u_t \,+\, f(u)_x \,=\, 0\ . \tag{2.25}$$

As in a MUSCL approximation, a mixed finite-difference/finite-volume method is used for the discretization in space. Let x_j , $1 \le j \le N$ denote the discretization points of the mesh. For each discretization point, we state : $u_j \approx u(x_j)$ and we define the control volume C_j as the interval $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, with $x_{j+\frac{1}{2}} = \frac{x_j+x_{j+1}}{2}$.

As in a finite-difference method, we define the unknown vector $U = \{u_j\}$ as point approximation values of the function $u(x)$ in each node j of the mesh. The time advancing is written:

$$U_{j,t} + \Psi_j(U) = 0 \tag{2.26}$$

where, similarly to finite-volumes, the vector $\Psi(U)$ is built according to approximations of $f(u)$ defined at cell boundaries:

$$\Psi_j(U) = \frac{1}{\Delta x}(\Phi_{j+\frac{1}{2}} - \Phi_{j-\frac{1}{2}}) \ ; \ \Phi_{j+\frac{1}{2}} = \Phi(u_j, u_{j+1}, f_{j+\frac{1}{2}^-}, f_{j+\frac{1}{2}^+}) \tag{2.27}$$

where

$$\Phi(u,v,f,g) = \frac{1}{2}\left[\,(1+\delta sign(c))f \ + \ (1-\delta sign(c))g\,\right] \ , \quad c = f'(u/2 + v/2).$$

Let us write down a particular flux:

$$\Phi_{j+\frac{1}{2}} = \Phi(u_j, u_{j+1}, f_{j+\frac{1}{2}^-}, f_{j+\frac{1}{2}^+}) = \frac{f_{j+\frac{1}{2}^-} + f_{j+\frac{1}{2}^+}}{2} + \delta sign(c)\frac{f_{j+\frac{1}{2}^-} - f_{j+\frac{1}{2}^+}}{2} \tag{2.28}$$

The coefficient δ controls the spatial dissipation. For defining the integration values $f_{j\pm\frac{1}{2}^\pm}$ of f at boundaries of control volume C_j, we apply the MUSCL methodology [Leer 1977], to the left and right control volume boundary fluxes; $f_{j\pm\frac{1}{2}^\pm}$ is built using linear interpolation formulas :

$$\begin{array}{lll} f_{j+\frac{1}{2}^-} = f_j + \frac{1}{2}\Delta f_{j+\frac{1}{2}^-} & ; & f_{j+\frac{1}{2}^+} = f_{j+1} - \frac{1}{2}\Delta f_{j+\frac{1}{2}^+} \\ f_{j-\frac{1}{2}^-} = f_{j-1} + \frac{1}{2}\Delta f_{j-\frac{1}{2}^-} & ; & f_{j-\frac{1}{2}^+} = f_j - \frac{1}{2}\Delta f_{j-\frac{1}{2}^+} \end{array}$$

where $\Delta f_{j\pm\frac{1}{2}^\pm}$ are slopes, i.e. approximations of difference term $\frac{\partial f}{\partial x}\Delta x$:

$$\begin{aligned} \Delta f_{j+\frac{1}{2}^-} &= (1-\beta)\,(f_{j+1} - f_j) + \beta\,(f_j - f_{j-1}) \\ \Delta f_{j+\frac{1}{2}^+} &= (1-\beta)\,(f_{j+1} - f_j) + \beta\,(f_{j+2} - f_{j+1})\ . \end{aligned} \tag{2.29}$$

We observe that if the fluxes are based on polynomial reconstruction from the *average* values of the unknowns, as in ENO finite-volumes, choosing a higher-order reconstruction will produces a higher-order finite-volume scheme. In the present *vertex-centered* context, *high-order accuracy is not obtained by a higher-order interpolation* but by an interpolation that compensates the error coming from the final central differencing in (2.27). This writes as follows:

$$\begin{aligned} \Delta f_{j+\frac{1}{2}^-} &= (1-\beta)\,(f_{j+1} - f_j) + \beta\,(f_j - f_{j-1}) \\ &\quad + \theta^c\,(\,-f_{j-1} + 3f_j - 3f_{j+1} + f_{j+2}\,) \\ &\quad + \theta^d\,(\,-f_{j-2} + 3f_{j-1} - 3f_j + f_{j+1}\,) \\ \Delta f_{j+\frac{1}{2}^+} &= (1-\beta)\,(f_{j+1} - f_j) + \beta\,(f_{j+2} - f_{j+1}) \\ &\quad + \theta^c\,(\,-f_{j-1} + 3f_j - 3f_{j+1} + f_{j+2}\,) \\ &\quad + \theta^d\,(\,-f_j + 3f_{j+1} - 3f_{j+2} + f_{j+3}\,) \end{aligned} \tag{2.30}$$

Schemes	δ	β	θ^c	θ^d	**Order**
1	1	1/3	0	0	3
2	1	1/3	- 1/6	0	4
3	1	1/3	0	- 1/6	4
4	1	1/3	- 1/10	- 1/15	5
4'	0	1/3	- 1/10	- 1/15	6

Table 2.1: Accuracy of different versions of the V6 schemes in 1D case.

where θ^c and θ^d are parameters that control the combination of fully upwind and centered corrections. In order to analyse more simply this scheme, we assume that

$$c = f'(u) \text{ is constant and equal to 1 .}$$

Then:

$$\begin{array}{rrcl}
\Psi_j(U) = \dfrac{1}{4\Delta x}(& & (1+\delta)\theta^d & f_{j-3} \\
+ & & [(1+\delta)\beta + 2\delta\theta^c - 4\delta\theta^d - 4\theta^d] & f_{j-2} \\
+ & & [-2(\beta+2\delta\beta+1) - 8\delta\theta^c + 5\theta^d + 7\delta\theta^d] & f_{j-1} \\
+ & & [6\delta\beta + 12\delta\theta^c - 8\delta\theta^d] & f_j \\
+ & & [2(\beta-2\delta\beta+1) - 8\delta\theta^c - 5\theta^d + 7\delta\theta^d] & f_{j+1} \\
+ & & [-(1-\delta)\beta + 2\delta\theta^c - 4\delta\theta^d + 4\theta^d] & f_{j+2} \\
 & & -(1-\delta)\theta^d & f_{j+3})
\end{array} \tag{2.31}$$

which gives:

$$\begin{aligned}
\Psi_j(U) \quad = \quad & \frac{\partial f}{\partial x} + C_3\frac{\Delta x^2}{6}\frac{\partial^3 f}{\partial x^3} + C_4\frac{\Delta x^3}{4}\frac{\partial^4 f}{\partial x^4} \\
& + C_5\frac{\Delta x^4}{120}\frac{\partial^5 f}{\partial x^5} + C_6\frac{\Delta x^5}{24}\frac{\partial^6 f}{\partial x^6} + O(\Delta x^6)
\end{aligned} \tag{2.32}$$

where:

$$\begin{aligned}
C_3 &= 1 - 3\beta \\
C_4 &= \delta(\beta + 2\theta^c + 2\theta^d) \\
C_5 &= 1 - 15\beta - 60\theta^d \\
C_6 &= \delta(\beta + 2\theta^c + 8\theta^d) \quad .
\end{aligned}$$

We observe that schemes described in (2.27)-(2.30) are in general second-order accurate but they become high-order accurate for some values of the parameters β, δ, θ^c and θ^d. Fifth-order accuracy is obtained with an adequate choice of the three coefficients, viz. $\beta = 1/3, \theta_c = -1/10, \theta^d = -1/15$. In that case, the numerical dissipation takes the form:

$$D_j(U) = \frac{\delta}{60\Delta x}(-f_{j-3} + 6f_{j-2} - 15f_{j-1} + 20f_j - 15f_{j+1} + 6f_{j+2} + f_{j+3}\ ). \tag{2.33}$$

The corresponding dissipative flux writes:

$$\bar{D}_{j+\frac{1}{2}}(U) = \frac{\delta}{60}(-f_{j-2} + 5f_{j-1} - 10f_j + 10f_{j+1} - 5f_{j+2} + f_{j+3}\ ). \tag{2.34}$$

Since this dissipation term is based on an upwinding paradigm, it has good explicit linear stability (analysed in the sequel), together with a balanced spatial stabilisation property. Putting $\delta = 0$, leads to a central-differenced (non-dissipative) sixth-order accurate scheme.

2.3.2.2 Time advancing stability

We can combine the above scheme with the standard Runge-Kutta time advancing.

$$\begin{aligned}
&U^{(0)} = U^n \\
&V_1 = \Delta t\ \Psi(U^n) \\
&V_2 = \Delta t\ \Psi(U^n + V_1/2) \\
&V_3 = \Delta t\ \Psi(U^n + V_2/2) \\
&V_4 = \Delta t\ \Psi(U^n + V_3) \\
&U^{n+1} \ = \ U^n + V_1/6 + V_2/3 + V_3/3 + V_4/6
\end{aligned} \tag{2.35}$$

In many cases a linearized version can be used. Let us recall the Jameson variant [Jameson 1993] which writes as follows (N-stage version):

$$\begin{aligned}
&U^{(0)} = U^n \\
&U^{(k)} = U^{(0)} + \frac{\Delta t}{N-k+1}\Psi\left(U^{(k-1)}\right), \quad k = 1 \ldots N \\
&U^{n+1} = U^{(N)}.
\end{aligned} \tag{2.36}$$

An A-stability analysis as in [Hirsch 1988] can be applied. We give in Table 2.2 some typical maximal CFL numbers for the six-stage RK scheme, which ensure a global accuracy order of five for the two best schemes of the proposed family. This table illustrates that the above schemes can be used with CFL number of the order of the unity.

All of these schemes can be advanced in time with implicit schemes such as BDF1 and BDF2, see [Debiez 1996]. Combination with unsteady Defect Correction [Martin 1996] is also possible. Linear stability is unconditional in all cases.

2.3.3 Unstructured two-dimensional case

In order to increase the accuracy of the second-order MUSCL construction, we introduce an enriched method for computing the extrapolation slopes $(\nabla f)_{ij}$ and

β	θ^c	θ^d	δ	CFL_{max}
			1	1
1/3	0	0	1	2.310
1/3	- 1/6	0	1	0.263
1/3	0	- 1/6	1	1.332
1/3	- 1/10	- 1/15	1	1.867
0	0	0	1	.303

Table 2.2: Maximal Courant numbers (explicit RK6 scheme) for the different LV6 spatial schemes (1D analysis).

$(\nabla f)_{ji}$:

$$\begin{aligned}(\nabla f)_{ij}.\vec{ij} = \; & (1-\beta)(\nabla f)^c_{ij}.\vec{ij} + \beta(\nabla f)^u_{ij}.\vec{ij} \\ & +\xi_c\left[(\nabla f)^u_{ij}.\vec{ij} - 2(\nabla f)^c_{ij}.\vec{ij} + (\nabla f)^d_{ij}.\vec{ij}\right] \\ & +\xi_d\left[(\nabla f)_{D^*_{ij}}.\vec{ij} - 2(\nabla f)_i.\vec{ij} + (\nabla f)_j.\vec{ij}\right] \quad ,\end{aligned} \tag{2.37}$$

The computation of f_{ji} is analogous:

$$\begin{aligned}(\nabla f)_{ji}.\vec{ij} = \; & (1-\beta)(\nabla f)^c_{ij}.\vec{ij} + \beta(\nabla f)^d_{ij}.\vec{ij} \\ & +\xi_c\left[(\nabla f)^u_{ij}.\vec{ij} - 2(\nabla f)^c_{ij}.\vec{ij} + (\nabla f)^d_{ij}.\vec{ij}\right] \\ & +\xi_d\left[(\nabla f)_{D^*_{ji}}.\vec{ij} - 2(\nabla f)_j.\vec{ij} + (\nabla f)_i.\vec{ij}\right] \quad ,\end{aligned} \tag{2.38}$$

The term $(\nabla f)_{D^*_{ij}}$ is the gradient at the point D^*_{ij}. As shown in Figure 2.4 D^*_{ij} is the intersection point between maxi-line S_iS_j with edge (face for 3D) nm of the upwind triangle. This last gradient is computed by interpolation of the nodal gradient values at the nodes contained in the edge opposite to i in the upwind triangle T_{ij}.

$$(\nabla f)_{D^*_{ij}} \;=\; \alpha(\nabla f)_m + (1-\alpha)(\nabla f)_n \;\text{ with }\; D^*_{ij} = \alpha m + (1-\alpha)n.$$

The coefficients ξ^c and ξ^d are upwinding parameters that control the combination of fully upwind and centered slopes.

Analysis of Cartesian case (2D/median/2D/circumcenter). We restrict to an advection model:

$$U_t + aU_x + bU_y = 0 \; . \tag{2.39}$$

The proposed schemes may have only sixth-order dissipation and are in general second-order accurate but they become higher-order accurate for some values of the parameters β (see [Desideri 1987]), ξ^c and ξ^d, see [Carpentier 1995] and Table 2.3.

	β	ξ^c	ξ^d	δ	Order
β-scheme	1/3	0	0	1	3
β-scheme	1/3	0	0	0	4
NLV6 Method	1/3	-1/30	-2/15	1	5

Table 2.3: Accuracy of different versions of the presented scheme in 2D Cartesian case.

For the case of unstructured meshes we can show only first-order accuracy in general, and second-order for smooth variation of mesh size. Better convergence can be observed in practice, we refer to [Abalakin 2002a]. Also, the level of dissipation of this family of schemes is much smaller than for usual MUSCL schemes, see [Debiez 1999].

2.3.4 High-order LV6 spatial scheme

We present now a more simple family of scheme, the linear V6 schemes, **LV6**, introduced in [Debiez 1996, Debiez 1999] in which *interpolation is applied to the primitive variables*. These schemes are built as follows:

0. A background flow $W = (\rho, \rho u, \rho v, E)$ on each vertex of the mesh is given.

1. Compute the primitive variable $U = (\rho, u, v, p)$ on each vertex (vertexwise loop).

2. Compute the nodal gradients ∇U :

$$(\nabla U)_i = \frac{1}{meas(C_i)} \sum_{T \in C_i} \frac{meas(T)}{3} \sum_{k \in T} U_k \nabla \Phi_k^T. \tag{2.40}$$

3. Start *edgewise assembly loop*:

- Compute the extrapolated slopes :

$$\begin{aligned}(\nabla U)_{ij}^{HO} \cdot \vec{ij} = \; & (1-\beta)(\nabla U))_{ij}^{c} \cdot \vec{ij} + \beta(\nabla U))_{ij}^{u} \cdot \vec{ij} \\ & +\xi_c \left[(\nabla U))_{ij}^{u} \cdot \vec{ij} - 2(\nabla U))_{ij}^{c} \cdot \vec{ij} + (\nabla U))_{ij}^{d} \cdot \vec{ij}\right] \\ & +\xi_d \left[(\nabla U))_{D_{ij}^*} \cdot \vec{ij} - 2(\nabla U))_i \cdot \vec{ij} + (\nabla U))_j \cdot \vec{ij}\right] \quad ,\end{aligned} \tag{2.41}$$

  and analogous for $(\nabla U)_{ji}^{HO}$.

- Define left and right variable interpolations:

$$\begin{aligned} U_{ij} &= U_i + \nabla U_{ij} \\ U_{ji} &= U_j - \nabla U_{ji} \end{aligned}$$

  and recover the left and right values of conservative variables U_{ij}, U_{ji}.

The upwind differenced flux then writes:

$$\Phi_{ij} = \Phi^{Riemann}(U_{ij}, U_{ji}) \tag{2.42}$$

and add (substract) it to flux assembly at vertex i (j) and multiply flux assembly by the inverse mass matrix in order to obtain the update of the variable.

Remark 1
As it, the scheme is not monotone and so needs to be limited to guarantee the TVD (Total Variation Diminishing) property of the scheme. The considered limiter is a generalisation of the Superbee limiter with three entries:

$$\begin{cases} Lim(\nabla^c, \nabla^d, \nabla^{HO}) = 0 \quad if \nabla^c \nabla^d \leq 0, \\ Lim(\nabla^c, \nabla^d, \nabla^{HO}) = Sign(\nabla^c) min\left(2|\nabla^c|, 2|\nabla^d|, |\nabla^{HO}|\right) \quad else. \end{cases}$$

Thus, for each variable of conservative vector W, for example the first component we have:

$$(\rho_{ij})_k = Lim(\nabla\rho^c_{ij}, \nabla\rho^d_{ij}, \nabla\rho^{HO}_{ij}).$$

where ρ^c_{ij} and ρ^d_{ij} and $\nabla\rho^{HO}_{ij}$ are defined by Relations (2.20) and respectively (2.21).

Remark 2
The combination of median cells and upwinding produces inconsistent error terms under the form of quotients $\frac{\Delta x}{\Delta y}$ or $\frac{\Delta y}{\Delta x}$ which can produce large errors when the previous quotients are large, that is when mesh is stretched. One way to escape this inconsistency without losing the low dispersion properties (typically, fifth-order accuracy) consists of using the Barth construction of cells. □

2.3.5 Time advancing

For explicit time advancing, we can use, as in the 1D case, a standart Runge-Kutta scheme or the linearized version defined above.
An implicit Backward-Differencing formula can also be applied with a spatially-first-order accurate simplified Jacobian. An interesting option is the second-order Backward-Differencing formula.

It can then be crucial to apply a multigrid iteration in combination with pseudo time-advancing (steady case) or (for both cases) an efficient implicit time advancing. Designing a multigrid scheme for unstructured meshes rises the problem of defining a series of coarser grids. In other words, we have to define several new meshes or to find an alternative strategy. In [Lallemand 1992, Francescatto 1998], this is done in a transparent manner from the fine mesh by using the so-called cell agglomeration. Parallel multigrid extensions are proposed in [Mavriplis 1997, Fournier 1998]. Concerning the implicit time-stepping, it needs a solution algorithms for at least a linearised problem. This can also be done with a multigrid algorithm. Another option well adapted to message passing parallelism is the Krylov-Newton-Schwarz (KNS) algorithm, as in [Knoll 2004]. A first version of KNS, under the form of the Restrictive Additive Schwarz **RAS** was developed in [Cai 1999, Sarkis 2000]. This method can produce second order convergence in space and time although using a spatially-first-order accurate simplified Jacobian. This is obtained by means of the two-step-Newton Defect Correction proposed by [Martin 1996].

2.3.6 Conclusion on superconvergent schemes

We have described a family of schemes for the Euler equations involving low numerical dissipation. Interested readers are refered to [Debiez 1996, Debiez 1999, Debiez 1998, Abalakin 2002a, Abalakin 2002b, Abalakin 2002c, Abalakin 2001, Abalakin 2004, Camarri 2001, Camarri 2002a, Camarri 2004, Gourvitch 2004].

Based on MUSCL schemes, the LV6 schemes involve sophisticated primitive variable reconstruction designed in order to enjoy low dissipation properties thanks to a model of sixth derivative. In the case of advection with uniform velocity, those schemes present superconvergence properties, in the sense that, when applied to a cartesian sub-region of the mesh, these second-order schemes are of higher order (up to sixth-order).

The accuracy order on a regular enough mesh is at least two. On very irregular meshes, regularity can slowly degrade. This issue is addressed in next section.

2.4 A new mass matrix formulation

2.4.1 Introduction

In the previous section, a MUSCL and superconvergent MUSCL schemes have been introduced. However, the accuracy of these schemes for unsteady calculations is obtained only if the mesh is smooth enough. In the case of very irregular meshes, accuracy can degrade to first order. Indeed, the control of the dispersive property of these schemes relies on the upwind formulation which, for regular meshes at least, compensate the dispersion introduced by the diagonal mass matrix. According to the finite element theory and practice (see for example [Belytschko 1978]), the consistent mass matrix has good dispersive properties when use in combination with central differencing. Higher accuracy order is obtained on regular mesh and better accuracy is observed in practice (see for example [Donea 1987]). It is therefore interesting to compare the mass lumped-upwind technology with consistent mass-artificial diffusion one.

In this section we introduce and analyse a mass matrix formulation (in the sense of finite element matrix) which intends to improve dispersion on irregular mesh, possibly with the same qualities in spatial stability, in particular for Large Eddy Simulation. The issue of an adequate artificial viscosity needs then to be carefully addressed.

The present section is organised as follows: we review in Section 2.4.2 a mass matrix scheme for the 1D advection equation and we analyse the stability of this scheme for explicit and implicit time advancing using Von Neumann analysis. We conclude with numerical experiments for a basic example of a Gaussian function travelling on a regular and respectively irregular mesh.

An application to Large Eddy Simulation (LES) of the scheme to be presented here is given in Chapter 3. This part of my thesis among with applications presented in Chapter 3 it is also presented as a research rapport co-authorired with Hilde Ouvrard and available online at $http://hal.inria.fr/inria-00429062_v1$.

2.4.2 Mass Matrix Central formulation for the advection equation

Spatial 1D MUSCL formulation: Let us review the one-dimensional scalar conservation law:

$$u_t + f(u)_x = 0. \tag{2.43}$$

We suppose that $f \in C^1(\mathbb{R}, \mathbb{R})$. We shall insist on particular cases where $f(u) = cu$ $(c > 0)$.

Spatial discretisation. Let us first consider a finite-volume method for the discretization in space. Let x_j , $1 \leq j \leq N$ denote the discretization points of the mesh. For each discretization point, we state : $u_j \approx u(x_j)$ and we define the control cell C_j as the interval $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ where $x_{j+\frac{1}{2}} = \frac{x_j + x_{j+1}}{2}$. We define the unknown vector $U = \{u_j\}$ as point approximation values of the function $u(x)$ in each node j of the mesh.
The semi-discrete formulation of Equation (2.43) writes:

$$U_{j,t} + \Psi_j(U) = 0 \tag{2.44}$$

where the vector $\Psi_j(U)$ is built according to approximations of $(f(u))_x$ defined at cell boundaries, as follows:

$$\Psi_j(U) = \frac{1}{\Delta x}(\Phi_{j+\frac{1}{2}} - \Phi_{j-\frac{1}{2}}) \text{ with } \Phi_{j+\frac{1}{2}} = \Phi(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}}), \tag{2.45}$$

where $u^-_{j\pm\frac{1}{2}}$, $u^+_{j\pm\frac{1}{2}}$ are integration values of u at boundaries of control volume C_j and Φ is a numerical flux function defined here :

$$\Phi(u, v) = \frac{cu + cv}{2} - \frac{\delta}{2}c(v - u), \tag{2.46}$$

where the coefficient δ controls the spatial dissipation.
For the upwind scheme (made of a dissipation based on sixth-order spatial derivatives) developed in Debiez *et al.*, see [Debiez 1998], and recalled previously in 2.3.4, these reconstructed values are given by: $u^-_{j+\frac{1}{2}} = u_j + \frac{1}{2}\Delta u^-_{j+\frac{1}{2}}$ and $u^+_{j+\frac{1}{2}} = u_j - \frac{1}{2}\Delta u^+_{j+\frac{1}{2}}$ (same reconstruction for $u^-_{j-\frac{1}{2}}$ and $u^+_{j-\frac{1}{2}}$) where the slopes $\Delta u^-_{j+\frac{1}{2}}$ and $\Delta u^+_{j+\frac{1}{2}}$ are defined for the LV6 scheme by:

$$\begin{aligned}\Delta u^-_{j+\frac{1}{2}} = {} & (1-\beta)(u_{j-1} - u_j) + \beta(u_j - u_{j-1}) \\ & + \theta^c(-u_{j-1} + 3u_j - 3u_{j+1} + u_{j+2}) \\ & + \theta^d(-u_{j-2} + 3u_{j-1} - 3u_j + u_{j+1})\end{aligned} \tag{2.47}$$

and

$$\begin{aligned}\Delta u^{+}_{j+\frac{1}{2}} &= (1-\beta)(u_{j+1}-u_j)+\beta(u_{j+2}-u_{j+1})\\ &+\theta^c(-u_{j-1}+3u_j-3u_{j+1}+u_{j+2})\\ &+\theta^d(-u_j+3u_{j+1}-3u_{j+2}+u_{j+3}) \qquad (2.48)\end{aligned}$$

With $\beta=\frac{1}{3}$, $\theta^c=-\frac{1}{10}$ and $\theta^d=-\frac{1}{15}$, this scheme becomes fifth-order accurate. The numerical dissipation introduced by this scheme is made of sixth-order derivatives and can be written as:

$$D_j(u)=\frac{D_{j+\frac{1}{2}}(u)-D_{j-\frac{1}{2}}(u)}{\Delta x},$$

where $D_{j+\frac{1}{2}}(u)=\frac{\delta c}{60}(-u_{j-2}+5u_{j-1}-10u_j+10u_{j+1}-5u_{j+2}+u_{j+3})$.

One important drawback of this scheme is that accuracy can deteriorate to first order for a very regular mesh. We depict the impact of this behaviour at the end of this chapter. Another drawback is a poor accuracy for stretched meshes, this last problem can be addressed by changing the shape of finite-volume cells.

Mass Matrix Scheme with central differencing: The usual central-differences three-point scheme is penalized by a dispersion leading error. This error can be compensated in the case where we introduce the finite-element P_1 consistent mass matrix, denoted here by M. That is, in temporal term of Equation (2.43) is evaluated by Finite-Element method. System (2.43) then becomes:

$$MU(t)\;+\;\Delta x\Psi(U)\;=\;0. \qquad (2.49)$$

The Finite-Element P_1 mass matrix is defined by: $(M)_{ij}=\int\Phi^{(i)}\Phi^{(j)}$ where $\Phi^{(i)}$ is the basis function for Finite-Element P_1 formulation. $\Phi^{(i)}$ is a piecewise linear function such that $\Phi^{(i)}(S_i)=1$ and $\Phi^{(i)}(S_j)=0$ for all $i\neq j$, where S_j denotes the j-th vertex. M is a symmetric three-diagonal matrix equal to:

$$M=\text{Three-diag}(\frac{1}{6}\Delta x,\frac{2}{3}\Delta x,\frac{1}{6}\Delta x)$$

One obtains thus for the j-th component of vector $MU(t)$:

$$(MU(t))_j\;=\;m_{j,j-1}U_{j-1,t}+m_{j,j}U_{j,t}+m_{j,j+1}U_{j+1,t}$$

where

$$m_{j,j-1}=1/6\Delta x,\;m_{j,j}=2/3\Delta x,\;m_{j,j+1}=1/6\Delta x. \qquad (2.50)$$

Then, we combine this discretisation of the temporal term with a numerical flux for the approximation of the convective term:

$$\Phi(u_j,u_{j+1})=c\left(\frac{u_j+u_{j+1}}{2}\right)-\frac{\delta}{2}T_{j+\frac{1}{2}}. \qquad (2.51)$$

Terms $c\left(\frac{u_j+u_{j+1}}{2}\right)$ will contribute to the central differencing flux.
We obtain a central differenced scheme that is fourth-order accurate on uniform and very regular meshes, but this scheme is unstable in most applications. We propose now to add a dissipation term buit in a similar way to the V6 Finite-Volume scheme of Section 2.3.2.
According to the Pascal triangle, a fifth-order difference evaluated between j and $j+1$ can be written as follows ($C > 0$):

$$T_{j+1/2} = Cc(-u_{j-2} + 5u_{j-1} - 10u_j + 10u_{j+1} - 5u_{j+2} + u_{j+3}). \tag{2.52}$$

We choose the constant C as:

$$C = \frac{\delta}{60}\ .$$

in order to have the same level of dissipation as in the upwind case . Finally, starting from expression (2.45) we get for the operator Ψ_j:

$$\begin{array}{rclll} \Psi_j(U) & = & \frac{c}{2\Delta x}(& C & u_{j-3} \\ & & + & (-6C) & u_{j-2} \\ & & + & (-1+15C) & u_{j-1} \\ & & + & (-20C) & u_j \\ & & + & (1+15)C & u_{j+1} \\ & & + & (-6C) & u_{j+2} \\ & & & C & u_{j+3} \\ & & & & &). \end{array} \tag{2.53}$$

Time advancing stability:

Explicit time stepping:

Let us consider a time integration of the system $MU_t = AU$, with A the spatial approximation matrix and M the P1 finite element mass matrix. We can combine the above scheme with the linearised six-stage Runge-Kutta scheme:

$$\begin{array}{l} U^{(0)} = U^n \\ U^{(k)} = U^{(0)} + \frac{\Delta t}{N-k+1} M^{-1}\Psi\left(U^{(k-1)}\right), \quad k = 1\ldots N \\ U^{n+1} = U^{(6)} \end{array} \tag{2.54}$$

The stability study of the scheme is made with the classical Fourier analysis. Let us include in equation (2.44) the Fourier mode: $\hat{u}_j^n = \overline{u}_k e^{ij\theta_k}$ where θ_k is the frequence parameter. We obtain:

$$m_\theta \frac{d\hat{u}_j^n}{dt} = -\hat{\Psi}^\delta$$

where $m_\theta = \frac{1}{3}(2 + \cos(\theta))$
$\hat{\Psi}^\delta = \frac{c}{2\Delta x}(R_3\cos(3\theta) + R_2\cos(2\theta) + R_1\cos(\theta) + R_0 + iI_1\sin(\theta))\hat{u}_j^n$, with:

$$\begin{cases} R_3 = 2\delta k \\ R_2 = -12\delta k \\ R_1 = 30\delta k \\ R_0 = -20\delta k \\ I_1 = 2 \end{cases}$$

denoting λ_θ the linear operator such as $\hat{\Psi}^\delta = -\lambda_\theta \hat{u}_j^n$.
We introduce the Courant number $\nu = \frac{c\Delta t}{\Delta x}$ and the amplification factor $G_\theta = g(z_\theta)$. $z_\theta = \frac{\lambda_\theta \Delta t}{m_\theta}$, g is the RK6 characteristic polynomial:

$$g(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + \frac{z^6}{720}$$

Finaly, we have:

$$\begin{cases} z_\theta = -\frac{3\nu}{2(2+\cos(\theta))}(z_\theta^R + i z_\theta^I) \\ z_\theta^R = R_3\cos(3\theta) + R_2\cos(2\theta) + R_1\cos(\theta) + R_0 \\ z_\theta^I = I_1 \sin(\theta) \end{cases}$$

Plotting the gain function $Ga(\nu) = \max\limits_{\theta\in[0,\pi]} g(z_\theta)$, we can determine ν_{max}, the maximum value of CFL number to obtain a stable scheme with, that is the maximum value of ν such as $|g(z_\theta)| \leq 1$.
For a CFL small enough (in practice we can take CFL=3.5), the whole of z_θ complex numbers remains inside the A-Stability region of the RK6 time advancing scheme as illustrated on Figure 2.6.

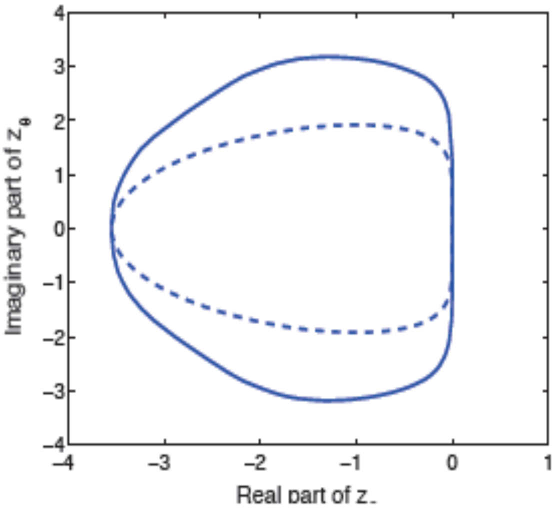


Figure 2.6: Stability analysis: we depict the boundary of the region for which $g(z) \leq 1$ and (inside) the eigenvalues of spatial operator

Implicit time stepping:

The purpose of this section is the stability analysis of the implicit scheme. This can be also done with Fourier analysis. The computation is made on the same one-dimensional scalar conservation law. Let us use the implicit scheme a δ-scheme using a mass-matrix formulation:

$$T^n \delta U^{n+1} = \Delta t^n \Psi(U^n)$$

with $\delta U^{n+1} = U^{n+1} - U^n$ and T^n is the implicit matrix.
In case of a first order scheme, T^n represents the following three-diagonal matrix :

$$T^n = \text{diag}(\tfrac{1}{6} - \nu, \tfrac{2}{3} + \nu, \tfrac{1}{6})$$

With the Fourier analysis we obtain:

$$t_\theta = 1 + \nu(1 - \cos(\theta)) + \tfrac{1}{3}(2 + cos(\theta)) + i\nu \sin(\theta)$$

The amplification factor is then given by:

$$G(\Delta t) = \tfrac{t_\theta + z_\theta}{t_\theta}$$

We are most interested on the behavior of the amplification factor when the time step Δt tends to $+\infty$. The aim is to know if the built ones schemes are preconditioned at first order with satisfactory factors of convergence. Let us denote $\lim\limits_{\Delta t \to \infty} G(\Delta t) = f_\theta(\delta)$. We are searching the Fourier's modes for different schemes witch maximises the functions:

$$f_\theta(\delta) = 1 - \frac{1}{4(1 - \cos(\theta))m_\theta} \left[(1 - \cos(\theta))z_\theta^R + \sin(\theta)z_\theta^I\right] + i \left[\frac{\sin(\theta)z_\theta^R - (1 - \cos(\theta))z_\theta^I}{4m_\theta(1 - \cos(\theta))}\right]$$

The functions f_δ are depicted with three different values of δ in Figures 2.7, 2.8 and 2.9. We can observe that our implicit scheme is inconditionally stable. Let us denote that the optimal value that minimise the gain function when Δt goes to ∞ is $\delta = 1$.

An extension of this formulation to three dimensional turbulent flows simulation is investigated in Chapter 3. The benchmarking of the classical VMS-LES approach and VMS-LES with the mass matrix formulation for simulating the flow around a circular cylinder at Reynolds number 3900 is presented in the same chapter.

2.4.3 Application: Gaussian translation on regular vs irregular mesh

In this section one computes a model problem of advection and we exhibit an irregular mesh for which the Finite-Volume methods behave very poorly and we

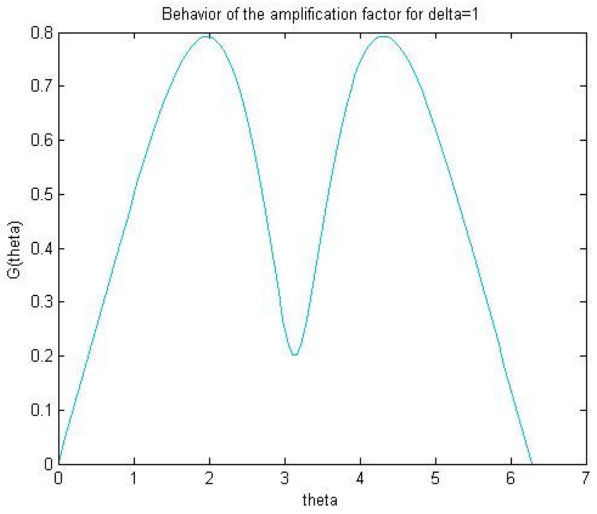


Figure 2.7: Gain function for $\delta = 1$

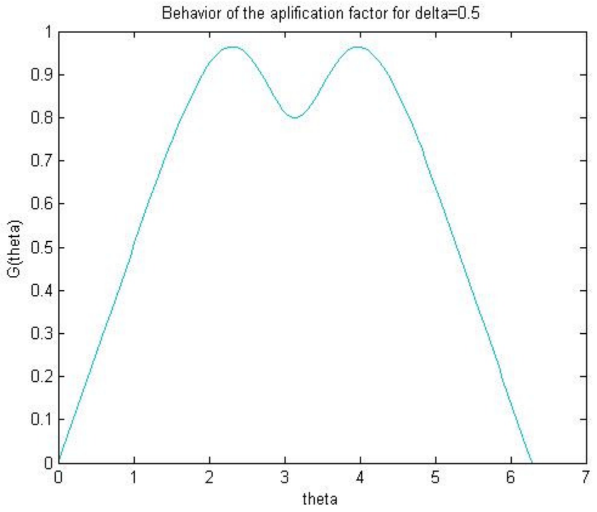


Figure 2.8: Gain function for $\delta = 0.5$

show the improvement carried by the novel method.

The graph of a Gaussian is a characteristic symmetric " bell shape curve " that quickly falls towards vanishing values.
For our 2D study, we consider the advection equation:

$$\frac{\partial \rho}{\partial t}(x, y, t) + c\frac{\partial \rho}{\partial x_i}(x, y, t) = 0$$

using an advection vector $c = (0, 1)$ and a Gaussian function as initial condition:

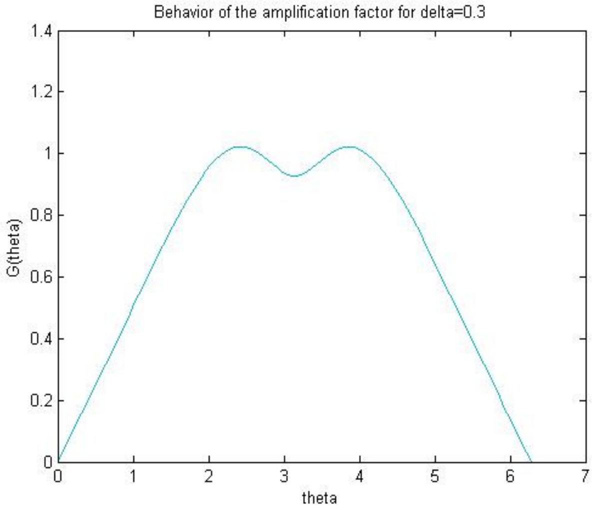


Figure 2.9: Gain function for $\delta = 0.3$

$$\rho(x, y, z, 0) = 1 + \exp^{-150(x+0,3)}$$

The advection equation is solved using the Mass Lumping V6 scheme and the Mass Matrix Central Differencing scheme with two options: projected dissipation and non projected dissipation. Recalling that a "Mass lumping" scheme means that the temporal term of the equations is treated by Finite Volume whereas Mass-Matrix scheme means a temporal term treated by Finite Element. We are in particular interested on the dissipation of the Gaussian when it is translated using different meshes. The first mesh is regular as shown Figure 2.10 and the second one is irregular with strong variations of the local mesh size (see Figure 2.13)

In the regular case, we plot respectively in Figure 2.12 and Figure 2.11 the Gaussian translation using the Mass Matrix Central Differencing scheme and the Mass Lumping V6 scheme. One can observe that the translation is well predicted. In the irregular case, as shown Figure 2.15, the translation is well predicted with the Mass Matrix Central scheme with a maximum deviation with respect to the exact solution of about 0.02 and using the V6 scheme, one have some perturbations with a maximum deviation of 0.15, more that 7 times larger (see Figure 2.14).

2.4.4 Preliminary conclusion

The new formulation need solving a mass matrix formulation even in the explicit time advancing. In this case the time step is 3 times smaller than the Finite-Volume method. In implicit time advancing, this drawback disappears. The first set of numerical experiments show an important improvement for very bad meshes. In Chapter 3 we give a second comparaison for turbulent flows.

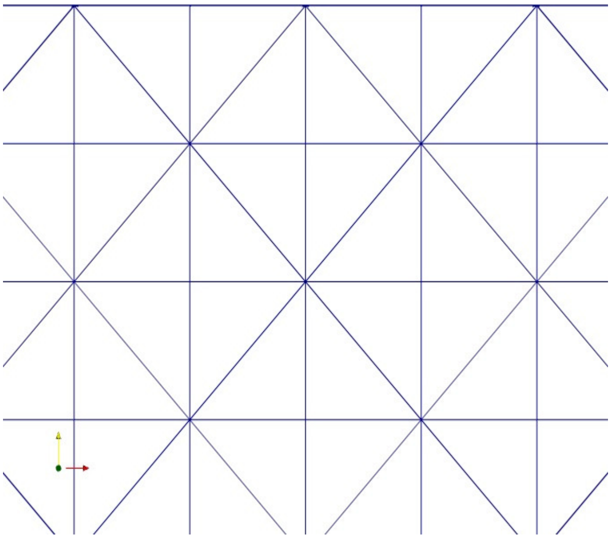


Figure 2.10: Regular mesh

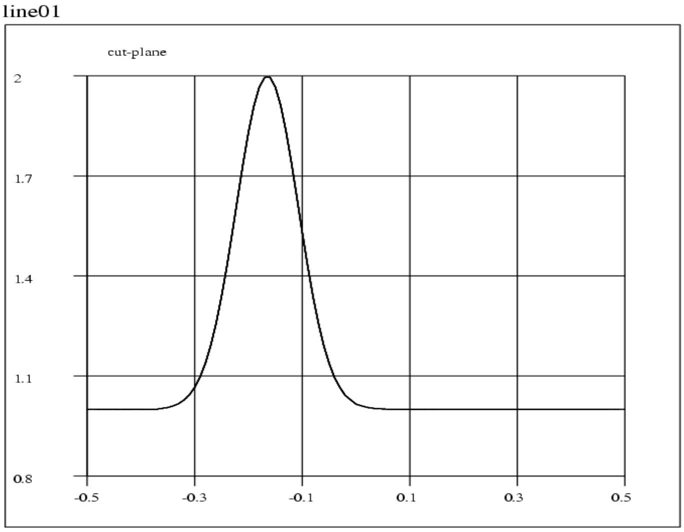


Figure 2.11: Gaussian translation for Mass Lumping V6 sheme on regular mesh

2.5 Numerical Method for 3D Navier-Stokes equations

2.5.1 Introduction

Remaining in the same framework of numerical schemes, but approaching to turbulent flows simulation, we introduce in the present section the numerical description and resolution of Navier-Stokes equations.
As it was explained for Euler flows, the spatial discretisation is based on a mixed finite element-volume formulation. Finite-volumes are used for the convective fluxes of Navier-Stokes equations and finite-elements (P1) for the diffusive ones.

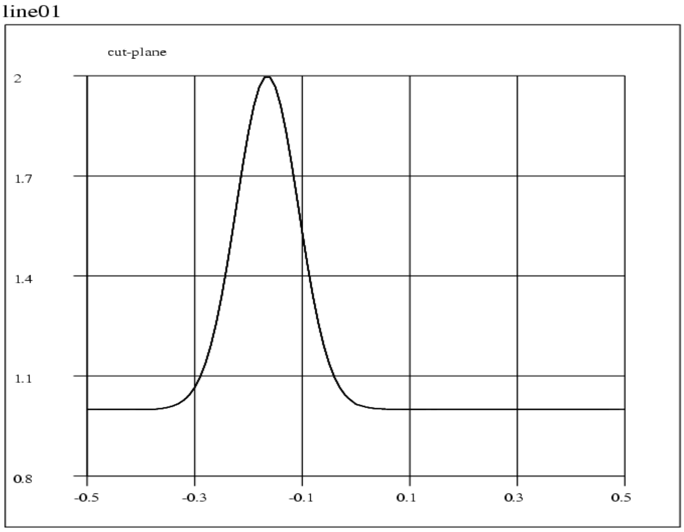


Figure 2.12: Gaussian translation for Mass Matrix Central Diff Projected scheme on regular mesh

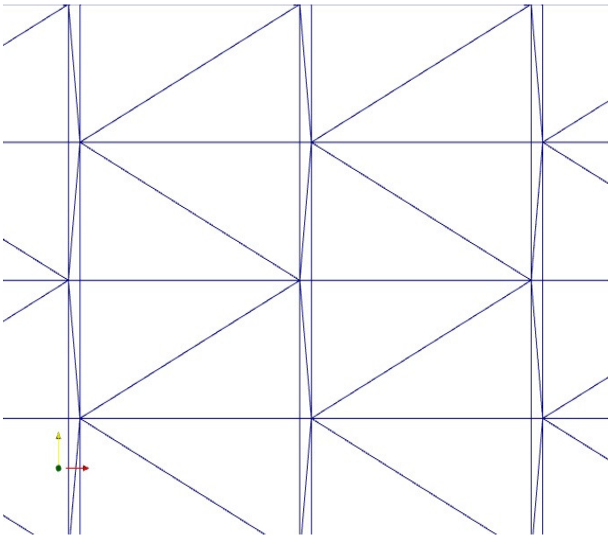


Figure 2.13: Irregular mesh

2.5.2 Set of equations

Suppose the Navier Stokes equations are numerically normalized with the following reference quantities:

- $L_{ref} \Longrightarrow$ characteristic length of the flow
- $U_{ref} \Longrightarrow$ velocity of the free-stream flow
- $\rho_{ref} \Longrightarrow$ density of the free-stream flow

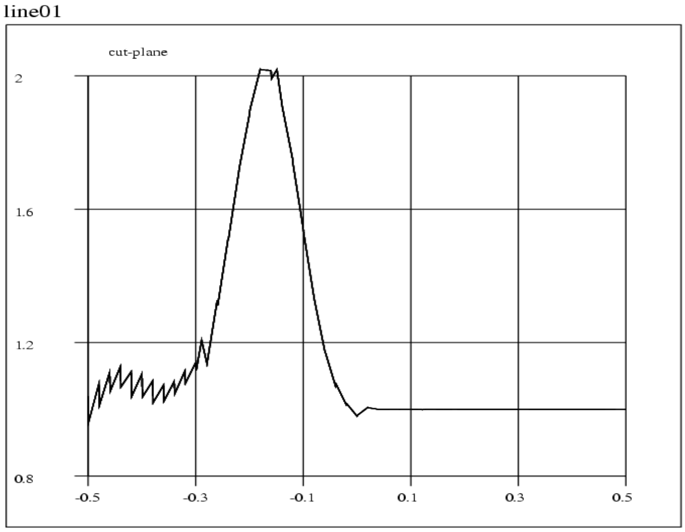


Figure 2.14: Gaussian translation for Mass Lumping V6 sheme on irregular mesh

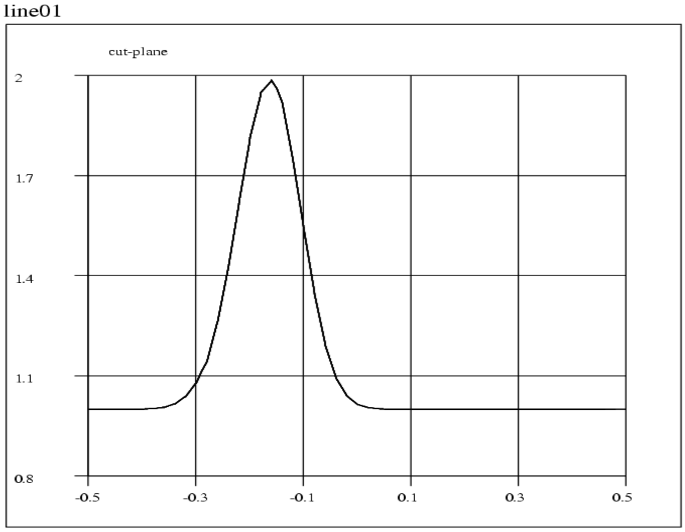


Figure 2.15: Gaussian translation for Mass Matrix Central Diff Projected scheme on irregular mesh

- $\mu_{ref} \Longrightarrow$ molecular viscosity of the free-stream flow

The flow variables can be normalised with the reference quantities as follows:

$$\rho^* = \frac{\rho}{\rho_{ref}} \qquad u_j^* = \frac{u_j}{U_{ref}} \qquad p^* = \frac{p}{p_{ref}}$$

$$E^* = \frac{E}{\rho_{ref} U_{ref}^2} \qquad \mu^* = \frac{\mu}{\mu_{ref}} \qquad t^* = t\,\frac{L_{ref}}{U_{ref}}\ . \tag{2.55}$$

The non-dimensional form of the Navier Stokes equations for the laminar case are reported in the following:

$$\begin{aligned}
\frac{\partial \rho^*}{\partial t^*} + \frac{\partial(\rho^* u_j^*)}{\partial x_j^*} &= 0 \\
\frac{\partial(\rho^* u_i^*)}{\partial t^*} + \frac{\partial \rho^* u_i^* u_j^*}{\partial x_j^*} &= -\frac{\partial p^*}{\partial x_i^*} + \frac{1}{Re}\frac{\partial \sigma_{ij}^*}{\partial x_j^*} \\
\frac{\partial(\rho^* E^*)}{\partial t^*} + \frac{\partial(\rho^* E^* u_j^*)}{\partial x_j^*} &= -\frac{\partial(p^* u_j^*)}{\partial x_j^*} + \frac{1}{Re}\frac{\partial(u_j^*\sigma_{ij}^*)}{\partial x_i^*} - \frac{\gamma}{RePr}\frac{\partial}{\partial x_j^*}\Big[\mu^*\Big(E^* - \frac{1}{2}u_j^* u_j^*\Big)\Big]
\end{aligned} \tag{2.56}$$

where the Reynolds number, $Re = U_{ref}L_{ref}/\nu$, is based on the references quantities, U_{ref} and L_{ref}, and on kinematic viscosity ν, the Prandlt number, Pr, can be assumed constant for a gas and equal to:

$$Pr = \frac{C_p \mu}{k}$$

with k the thermal conductivity, and $\gamma = C_p/C_v$ is the ratio between the specific heats at constant pressure and volume. Also the constitutive equations for the viscous stresses and the state equations may be written in non-dimensional form as follows:

$$\begin{aligned}
\sigma_{ij}^* &= -\frac{2}{3}\mu^*\Big(\frac{\partial u_k^*}{\partial x_k^*}\delta_{ij}\Big) + \mu^*\Big(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*}\Big) \\
p^* &= (\gamma - 1)\rho^*\Big(E^* - \frac{1}{2}u_j^* u_j^*\Big)\ .
\end{aligned} \tag{2.57}$$

In order to rewrite the governing equations in a compact form more suitable for the discrete formulation, the following unknown variables are grouped together in the $\boldsymbol{W}$ vector:
$\boldsymbol{W} = (\rho,\ \rho u,\ \rho v,\ \rho w,\ \rho E)^T$.
If two other vectors, $\boldsymbol{F}$ and $\boldsymbol{V}$ are defined as function of $\boldsymbol{W}$, as follows:

$$\boldsymbol{F} = \begin{pmatrix} \rho u & \rho v & \rho w \\ \rho u^2 + p & \rho uv & \rho uw \\ \rho uv & \rho v^2 + p & \rho vw \\ \rho uw & \rho vw & \rho w^2 + p \end{pmatrix}$$

and

$$\boldsymbol{V} = \begin{pmatrix} 0 & 0 & 0 \\ \sigma_{11} & \sigma_{21} & \sigma_{31} \\ \sigma_{12} & \sigma_{22} & \sigma_{32} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \\ u\sigma_{11} + v\sigma_{12} + w\sigma_{13} - q_1 & u\sigma_{12} + v\sigma_{22} + w\sigma_{23} - q_2 & u\sigma_{13} + v\sigma_{23} + w\sigma_{33} - q_3 \end{pmatrix}$$

they may be substituted in (2.56), to get a different compact format of the governing equations which is the starting point for the derivation of the Galerkin formulation and of the discretization of the problem:

$$\frac{\partial \boldsymbol{W}}{\partial t} + \frac{\partial}{\partial x_j} F_j(\boldsymbol{W}) - \frac{1}{Re}\frac{\partial}{\partial x_j} V_j(\boldsymbol{W}, \nabla \boldsymbol{W}) = 0 \ . \tag{2.58}$$

It is important to stress that the vectors $\boldsymbol{F}$ and $\boldsymbol{V}$ are respectively the convective fluxes and the diffusive fluxes.

2.5.3 Spatial discretization

Spatial discretization is based on a mixed finite-volume/finite-element formulation. A finite volume upwind formulation is used for the treatment of the convective fluxes while a classical Galerkin finite-element centred approximation is employed for the diffusive terms.
The computational domain Ω is approximated by a polygonal domain Ω_h. This polygonal domain is then divided in N_t tetrahedron T_i:

$$\Omega_h = \bigcup_{i=1}^{N_t} T_i. \tag{2.59}$$

The set of elements T_i forms the mesh used in the finite-element formulation. The dual finite-volume grid can be built starting from the triangulation following the medians method.

Regarding the convective fluxes we refer to the first part of this Chapter, more precisely to Section 2.3, where the second order MUSCL scheme is presented, which we recall it expresses the Roe flux as a function of the extrapolated values of W at the interface between two neighboring cells C_i and C_j, W_{ij} and W_{ji}.

2.5.4 Convective fluxes

Convective fluxes are defined either by the MUSCL LV6 scheme described in Section 2.3 or by the consistent central fluxes

2.5.5 Diffusive fluxes

The P1 finite-element basis function, $\phi^{(i,T)}$, restricted to the tetrahedron T is assumed to be of unit value on the node i and to vanish linearly at the remaining vertices of T. The Galerkin formulation for the diffusive terms is obtained by multiplying the diffusive terms by $\phi^{(i,T)}$ and integrating over the domain Ω_h:

$$\iint_{\Omega_h} \Big(\frac{\partial V_j}{\partial x_j}\Big) \phi^{(i,T)} \,\mathrm{d}\Omega = \iint_T \frac{\partial V_j}{\partial x_j} \phi^{(i,T)} \,\mathrm{d}\Omega \ . \tag{2.60}$$

Integrating by parts the right-hand side of Equation (2.60) we obtain:

$$\iint_T \frac{\partial V_j}{\partial x_j}\phi^{(i,T)}\,\mathrm{d}\Omega = \int_{\partial T} V_j \phi^{(i,T)} n_j \,\mathrm{d}\sigma - \iint_T V_j \frac{\partial \phi^{(i,T)}}{\partial x_j}\,\mathrm{d}\Omega \ . \tag{2.61}$$

In order to build the fluxes for the node i consistently with the finite-volume formulation, the contribution of all the elements having i as a vertex needs to be summed together as follows:

$$\begin{aligned} &\sum_{T,i\ \in\ T} \Big(\int_{\partial T} V_j \phi^{(i,T)} n_j \,\mathrm{d}\sigma - \iint_T V_j \frac{\partial \phi^{(i,T)}}{\partial x_j}\,\mathrm{d}\Omega \Big) = \\ -&\sum_{T,i\ \in\ T} \iint_T V_j \frac{\partial \phi^{(i,T)}}{\partial x_j}\,\mathrm{d}\Omega + \int_{\Gamma_h=\partial\Omega_h} \phi^{(i,T)} V_j n_j \,\mathrm{d}\sigma \ . \end{aligned} \tag{2.62}$$

In the P1 formulation for the finite-element method, the test functions, $\phi^{(i,T)}$, are linear functions on the element T and so their gradient is constant. Moreover, in the variational formulation the unknown variables contained in $\boldsymbol{W}$ are also approximated by their projection on the P1 basis function. For these reasons the integral can be evaluated directly.

2.5.6 Boundary conditions

Firstly, the real boundary Γ is approximated by a polygonal boundary Γ_h that can be split in two parts:

$$\Gamma_h = \Gamma_\infty \cup \Gamma_b \tag{2.63}$$

where the term Γ_∞ represents the far-fields boundary and Γ_b represents the body surface. The boundary conditions are set using the Steger-Warming formulation ([Steger 1981]) on Γ_∞ and using slip or no-slip conditions on Γ_b.

2.5.7 Time advancing

Once the equations have been discretized in space, the unknown of the problem is the solution vector at each node of the discretization as a function of time, $\boldsymbol{W}_h(t)$. Consequently the spatial discretization leads to a set of ordinary differential equations in time:

$$\frac{d\boldsymbol{W}_h}{dt} + \Psi(\boldsymbol{W}_h) = 0 \tag{2.64}$$

where Ψ_i is the total flux, containing both convective and diffusive terms, of W_h through the i-th cell boundary divided by the volume of the cell.

2.5.7.1 Explicit time advancing

In the explicit case a N-step low-storage Runge-Kutta algorithm is used for the discretization of Eq.(2.64):

$$\begin{cases} \boldsymbol{W}^{(0)} = \boldsymbol{W}^{(n)}, \\ \boldsymbol{W}^{(k)} = \boldsymbol{W}^{(0)} + \Delta t \ \alpha_k \ \Psi(\boldsymbol{W}^{(k-1)}), \quad k = 1, \dots, N \\ \boldsymbol{W}^{(n+1)} = \boldsymbol{W}^{(N)}. \end{cases}$$

in which the suffix h has been omitted for sake of simplicity. Different schemes can be obtained varying the number of steps, N, and the coefficients α_k.

2.5.7.2 Implicit time advancing

For the implicit time advancing scheme in AERO the following second order accurate backward difference scheme is used:

$$\alpha_{n+1}\boldsymbol{W}^{(n+1)} + \alpha_n \boldsymbol{W}^{(n)} + \alpha_{(n-1)}\boldsymbol{W}^{(n-1)} + \Delta t^{(n)}\Psi(\boldsymbol{W}^{(n+1)}) = 0 \tag{2.65}$$

where the coefficients α_n can be expressed as follows:

$$\alpha_{n+1} = \frac{1+2\tau}{1+\tau}, \qquad \alpha_n = -1-\tau, \qquad \alpha_{n-1} = \frac{\tau^2}{1+\tau} \tag{2.66}$$

where $\Delta t^{(n)}$ is the time step used at the n-th time iteration and

$$\tau = \frac{\Delta t^{(n)}}{\Delta t^{(n-1)}}. \tag{2.67}$$

The nonlinear system obtained can be linearised as follows:

$$\begin{aligned} \alpha_{n+1}\boldsymbol{W}^{(n)} + \alpha_n \boldsymbol{W}^{(n)} + \alpha_{(n-1)}\boldsymbol{W}^{(n-1)} + \Delta t^{(n)}\Psi(\boldsymbol{W}^{(n)}) = \\ -\Big[\alpha_{n+1} + \delta t^{(n)}\frac{\partial \Psi}{\partial \boldsymbol{W}}(\boldsymbol{W}^{(n)})\Big](\boldsymbol{W}^{(n+1)} - \boldsymbol{W}^{(n)}). \end{aligned} \tag{2.68}$$

Following the defect-correction approach, the jacobians are evaluated using the 1st order flux scheme (for the convective part), while the explicit fluxes are composed with 2nd order accuracy. The resulting linear system is solved by a Schwarz method.

2.6 Conclusion

We have recalled the feature of the numerical platform used in this thesis for investigation in turbulence modelling and in mesh adaptation methods. We have also proposed and analysed a new version of the numerical scheme, with its explicit and implicit version. On a simplified model, the improvement is important for irregular meshes. A second conclusion is presented for turbulence models in Chapter 3.

Chapter 3

Turbulence modelling

Contents

3.1 Introduction

In spite of extensive research for more than a century, modelling turbulent flows remains a big challenge nowadays. Turbulence simulation, in its purest form - called Direct Numerical Simulation or DNS - aims to resolve all details and scales of the turbulence, *i.e.* the *spatial* and *temporal* evolution of the entire range of eddies. To achieve this, Blazek [Blazek 2001] pointed out that a sufficient spatial resolution for DNS is proportional to $Re^{\frac{9}{4}}$ and the CPU time requirement is proportional to Re^3. In an attempt to resolve flow at Reynolds number 10000 using DNS, Tremblay

[F. Tremblay 2000] concludes that the simulation needs about 80 computing days running on a 32 processor parallel computer system. For practical flows around an aircraft the Reynolds number is measured in term of tens of million, that is -instead of 80 days, more that a thousand centuries would be necessary, even on one of the most powerfull supercomputer of nowadays who accounts for about 500000 processors (the Japanese supercomputer at $RIKEN$ Advance Institute for Computational Sciences, capable to perform more that 8 quadrillon operations per second accordingly to ISC events from june 2011). Thus, it is still not practical to accurately resolve the non-linear nature and three dimensional characteristics of turbulence at rather moderate and high Reynolds numbers using DNS with the currently available computer technology.
In fact the aircraft designer is not interested in all the details of the instantaneous flow but mostly in the statistical mean of the flow and a few bulk moments. To answer this, scientists and engineers focused on obtaining a model for the mean flow. The statistical simulation consists in ensemble-averaging the Navier-Stokes equations. This leads to the Reynolds Averaged Navier Stokes (RANS) models, where the apparent randomness and most part of, if not all, unsteadiness of the flow are averaged out, and most if not all aspects of turbulence are modelled. However, RANS models presents some degree of empiricism, making them less reliable in certain types of flow. LES is a kind of midway between DNS and RANS modelling as regards informations acquired on flow and the cost of calculating.

Large-Eddy Simulation (LES) is becoming an increasingly popular approach also for applications of industrial or engineering interest. It directly computes the large-scale turbulent structures which are responsible for an important part of the transfer of energy and momentum in a flow while modelling the smaller scales of dissipative and more isotropic structures. In order to distinguish between the large scales and small scales, a filter function is used in LES. The filter function dictates which eddies are large by introducing a length scale, the characteristic filter width of the simulation. All eddies larger than this length scale are resolved, while those smaller than the length scale are not. The effect of the small eddies eliminated by the operation of filtering on large ones is modelled by the so-called sub-grid scales (SGS) models. Although requiring much larger computational resources than RANS [1], LES has the potential to give more accurate results than the Reynolds-Averaged Navier-Stokes (RANS) approach for flows characterized by massive separation or significant unsteadiness. Paradigmatic examples of such flows are bluff-body flows.

A new approach to LES based on a variational multiscale (VMS) framework was recently introduced in Hughes *et al.* [T.J.R. Hughes 2000]. The main idea of VMS-LES is to decompose, through Galerkin projection, the resolved scales into

[1] LES computation requires a much finer mesh to resolve the smaller vortices, thus the computation becomes more expensive

larger and smaller ones and to add the SGS model only to the smallest scales. It aims at reducing the excessive dissipation introduced by eddy-viscosity SGS models on the large scales.

A bottle-neck for LES and VMS-LES models applied to high-Reynolds wall-bounded flows is that practical applications carried out now for more than two decades have highlighted that an extremely fine grid resolution, almost comparable to the one of direct numerical simulations, is needed in boundary-layers to obtain accurate results. Clearly, this leads to dramatic computational requirements for large Reynolds numbers. In this context, hybrid strategies have been proposed in the literature. They combine RANS and LES approaches together in order to obtain simulations as accurate as LES and at reasonable computational costs as RANS. Among the hybrid models described in the litterature, the Detached Eddy Simulation (DES) has received the largest attention [P.R. Spalart 1997]. This approach is generally based on the Spalart-Allmaras RANS model, modified in such a way that, far from solid walls and with refined grids, the simulation switches to the LES mode with a one-equation SGS closure. Another hybrid approach has been recently proposed, the Limited Numerical Scales (LNS) of [P. Batten 2000]), in which the blending parameter depends on the values of the eddy-viscosity given by a RANS model and of the SGS viscosity given by a LES closure. In practice, the minimum of the two eddy-viscosities is used. An example of validation of LNS for the simulation of bluff-body flows is given in [Camarri 2005].

A major difficulty in combining a standard RANS model with a LES one is due to the fact that RANS does not naturally allow for fluctuations, due to its tendency to damp them and therefore to "perpetuate itself", as explained in [P.R. Spalart 1997]. On the other hand, LES needs a significant level of fluctuations in order to model the flow accurately. The abrupt passage from a RANS region to a LES one may produce the so-called "modeled stress depletion" [P.R. Spalart 1997]. Nowadays, researchers try to address this issue by progressive blending.

We focus in this chapter on an hybrid model blending RANS and VMS-LES [Camarri 2005] and on VMS-LES both for non-dynamic and respectively dynamic SGS models.

This part of my thesis presents a joint work with Bruno Koobus and Hilde Ouvrard (both from University of Montpellier II, France), Maria-Vittoria Salvetti (University of Pisa, Italy), Stephen Wornom and Olivier Allain (both from Lemma Engineering company, France) and Alain Dervieux (INRIA Sophia Antipolis, France). Our team works on modelling turbulent flows through VMS-LES and hybrid RANS/LES approaches. One of the major motivation of our group is to develop a model able to handle computations of *mean or high Reynolds numbers* for *very complex geometries*, as the ones encountred in industrial applications. A typical academic configuration is represented by the flow around a circular cylinder of ideally infinite span, numerically modeled by using periodic boundary conditions on a spanwise width somewhere between 3 and 5 diameters. A very fine mesh is used. On the contrary, industrial studies can be confronted to much more difficult problems, as for instance an assembly of tens blunt bodies (think of an oil platform with its tanks and pastles as in Figure 3.1(top)) or long submarine cables (Figure 3.1 bottom), the length of which can be thousands times the diameter and that needs to be fully computed because of the deformation of the cable and of fluid flow velocity variation applied more or less in normal direction to the cable. Examples on Figure 3.1 are computed with the models developed by the Montpellier-INRIA-Pisa team. A more detailed discussion can be found in our paper [A. Belme 2010].

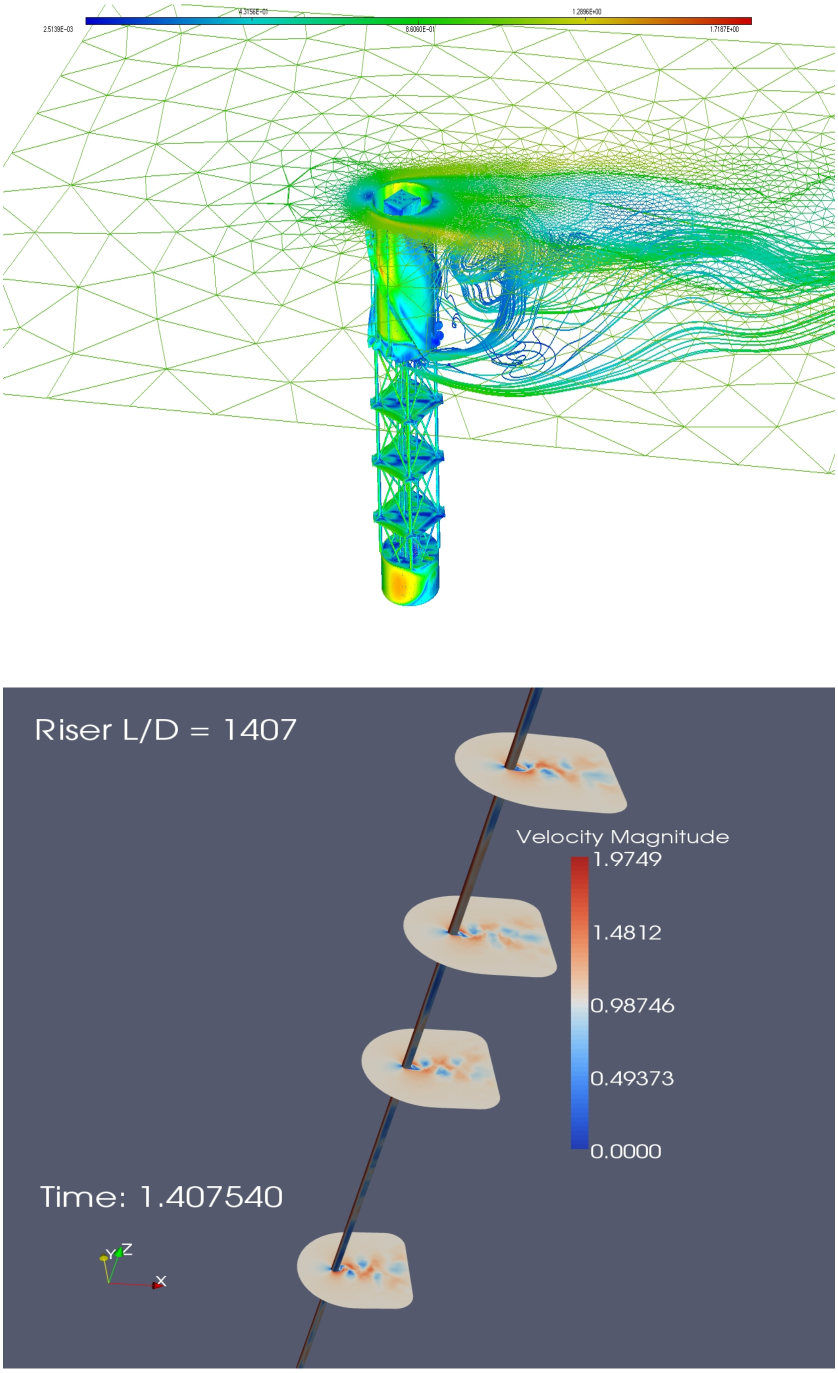


Figure 3.1: Examples of complex industrial applications: VMS-LES simulation of flow around a Spar-Truss (image on top) and around a riser (bottom image). A high Reynolds number characterizes the physics of the flow. We need to predict the main features of these unsteady flows without using a too large number of grid points, especially since other complex and costly effects need to be treated, as for instance the fluid-structure coupling. Computation perfomed by S. Wornom

The chapter is organized as follows: we start with an overview of the main features of classical LES modelling, which is useful for introducing the VMS-LES in Section 3.2.2. The SGS models of Smagorinsky, WALE and their dynamical VMS-LES counterparts are described in the same section. Then in Section 3.3 we give an overview of some hybrid models in the litterature (LNS, DES, DDES), followed by the description of our hybrid RANS/VMS-LES model. The benchmarking of our VMS-LES and of our hybrid RANS/VMS-LES strategies for the flow around a circular cylinder is presented in Sections 3.6 and 3.7.

Regarding the numerical methodology we employed in applications, we refer to Chapter 2.

The content of this chapter was presented in two conferences: ICAS 2010 [A. Belme 2010], AIMETA 2011 [Belme 2011].

3.2 Variational Multiscale approach for Large Eddy Simulation

3.2.1 Classical LES approach

Large Eddy Simulation (LES) is classified as a space filtering method in turbulence modelling, in order to get rid of the high frequency fluctuations. In LES, each variable W of the flow is separated into a filtered, resolved part $\overline{W}$ corresponding to eddies larger than the filter and a sub-filter unresolved part w', the scale of which is assumed to be too small for a good numerical representation on the considered mesh.

$$W = \overline{W} + w' \tag{3.1}$$

The filtered Navier-Stokes equations for compressible flows and in conservative form are considered. After separating the Navier-Stokes equation into resolved filtered and unresolved sub-filter components, unknown stress terms arise due to the nonlinearity of the equations and the shear stress of the flow. These terms need to be approximated to close the filtered Navier-Stokes equations. In modeling the subgrid-scales (SGS) terms resulting from filtering the Navier-Stokes equations, the effects of compressibility present in the SGS fluctuations are assumed to be low and the heat transfer and temperature gradients are assumed to be moderate. Thus, the retained SGS term in the momentum equations is the classical SGS stress tensor:

$$M_{ij} \; = \; \overline{\rho u_i u_j} \; - \; \overline{\rho}\tilde{u}_i\tilde{u}_j \quad , \tag{3.2}$$

where the over-line denotes the grid filter and the tilde the density-weighted Favre filter $(\tilde{f} = \left(\overline{\rho f}\right) / \left(\overline{\rho}\right))$. The isotropic part of M_{ij} can be neglected under the assumption of low compressibility effects in the SGS fluctuations. The deviatoric part, T_{ij}, is expressed by an eddy viscosity term:

$$T_{ij} = -2\mu_{\mathrm{sgs}} \left(\widetilde{S_{ij}} - \frac{1}{3}\widetilde{S_{kk}} \right) \quad , \tag{3.3}$$

where the resolved strain tensor $\widetilde{S_{ij}} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$ and μ_{sgs} can be, for example, one of the simplest SGS viscosity defined by the Smagorinsky SGS model:

$$\mu_{\mathrm{sgs}} = \bar{\rho}(C_s\Delta)^2|\widetilde{S}|$$

with Δ designing the filter width, C_s a constant that must be *a priori* assigned, and $\widetilde{S} = \sqrt{(2\widetilde{S_{ij}}\widetilde{S_{ij}})}$. Regarding the filter width, no reliable criterion exists to define the width of the filter corresponding to the numerical discretisation on unstructured grids. Nevertheless, the following expression has been employed here for each grid element T:

$$\Delta^{(T)} = Vols(is)^{1/3} \tag{3.4}$$

in which $Vols(is)$ is the volume associated to finite-volume cell is.

In the total energy equation, the effect of the SGS fluctuations has been modeled by the introduction of a constant SGS Prandtl number to be *a priori* assigned:

$$Pr_{\mathrm{sgs}} = C_p\frac{\mu_{\mathrm{sgs}}}{K_{\mathrm{sgs}}} \tag{3.5}$$

where K_{sgs} is the SGS conductivity coefficient; it takes into account the diffusion of total energy caused by the SGS fluctuations and is added to the molecular conductivity coefficient. We refer to [Camarri 1999] and [Camarri 2002b] for a more detailed discussion of the simplifying assumptions leading to the adopted SGS modeling in the unstructured case.

3.2.2 Variational Multiscale LES approach

The VMS-LES differs fundamentally from the traditional LES in a number of ways. In this new approach, one does not filter the Navier-Stokes equations but uses instead a variational projection relying on particular discretization basis functions. In particular, it models the effect of the unresolved-scales only in the equations representing the smallest resolved-scales, and not in the equations for the large scales. Consequently, in the VMS-LES, energy is extracted from the fine resolved-scales by a traditional model such as Smagorinsky eddy viscosity model, but no energy is directly extracted from the large structures in the flow. In particular, large structures are not directly diffused near a wall and one can reasonably hope to obtain a better behavior in boundary layers, and less dissipation in the presence of large coherent structures.

The initial development of the VMS-LES method historically focused on incompressible turbulent flows, regular grids, and spectral discretisations where the separation *a priori* of the scales is simple to achieve. For finite element approximations, a hierarchical basis approach and an alternative method based on cell agglomeration were proposed for *a priori* separating the coarse- and fine-scales [Koobus 2004, C. Farhat 2005, Ouvrard 2010]. In most cases, the VMS-LES method was applied mainly to homogeneous isotropic incompressible turbulence, and recently to incompressible turbulent channel flows, for which it has demonstrated an improvement over the traditional LES method [John 2006].

In this Variational Multiscale approach for Large Eddy Simulation (VMS-LES) the flow variables are decomposed as follows:

$$W_i = \underbrace{\overline{W}_i}_{LRS} + \underbrace{W_i'}_{SRS} + {W_i}^{SGS} \tag{3.6}$$

where $\overline{W}_i$ are the large resolved scales (LRS), W_i' are the small resolved scales (SRS) and ${W_i}^{SGS}$ are the unresolved scales. This decomposition is obtained by variational projection in the LRS and SRS spaces respectively. In the present study, we follow the VMS approach proposed in [Koobus 2004] for the simulation of compressible turbulent flows through a finite volume/finite element discretization on unstructured tetrahedral grids. If ψ_l are the N finite-volume basis functions and ϕ_l the N finite-element basis fonctions associated with the considered grid, in order to obtain the VMS flow decomposition in Equation (3.6), the finite dimensional spaces $\mathcal{V}_{FV}$ and $\mathcal{V}_{FE}$, respectively spanned by ψ_l and ϕ_l, can be in turn decomposed as follows [Koobus 2004]:

$$\mathcal{V}_{FV} = \overline{\mathcal{V}}_{FV} \bigoplus \mathcal{V}'_{FV} \ ; \quad \mathcal{V}_{FE} = \overline{\mathcal{V}}_{FE} \bigoplus \mathcal{V}'_{FE} \tag{3.7}$$

in which $\bigoplus$ denotes the direct sum and $\overline{\mathcal{V}}_{FV}$ and $\mathcal{V}'_{FV}$ are the finite volume spaces associated to the largest and smallest resolved scales, spanned by the basis functions $\overline{\psi_l}$ and $\psi_l^{'}$; $\overline{\mathcal{V}}_{FE}$ and $\mathcal{V}^{'}_{FE}$ are the finite element analogous. In [Koobus 2004] a projector operator P in the LRS space is defined by spatial average on macro cells in the following way:

$$\overline{\boldsymbol{W}} = P(\boldsymbol{W}) = \sum_k \underbrace{\left(\frac{Vol(C_k)}{\displaystyle\sum_{j\varepsilon I_k} Vol(C_j)} \sum_{j\varepsilon I_k} \psi_j \right)}_{\overline{\psi}_k} \boldsymbol{W}_k \tag{3.8}$$

for the convective terms, discretized by finite volumes, and:

$$\overline{\boldsymbol{W}} = P(\boldsymbol{W}) = \sum_k \underbrace{\left(\frac{Vol(C_k)}{\displaystyle\sum_{j\varepsilon I_k} Vol(C_j)} \sum_{j\varepsilon I_k} \phi_j \right)}_{\overline{\phi}_k} \boldsymbol{W}_k \tag{3.9}$$

for the diffusive terms, discretized by finite elements. In both Equations (3.8) and (3.9), $I_k = \{j | C_j \in C_{m(k)}\}$, $C_{m(k)}$ being the macro-cell containing the cell C_k (see

Figure 3.2). The macro-cells are obtained by a process known as agglomeration [Lallemand 1992]. The basis functions for the SRS space are clearly obtained as follows: $\psi_l^{'} = \psi_l - \overline{\psi}_l$ and $\phi_l^{'} = \phi_l - \overline{\phi}_l$.

A key feature of the VMS-LES approach is that the modeled influence of the unresolved scales on large resolved ones is set to zero, and so the SGS model is added only to the smallest resolved scales (which models the dissipative effect of the unresolved scales on small resolved ones). This leads to the following equations after semi-discretizations [Koobus 2004] :

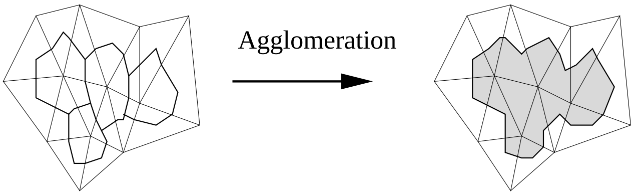


Figure 3.2: Construction of a macro-cell.

$$
\begin{aligned}
\int_{C_i} \frac{\partial \rho}{\partial t} \mathrm{d}\Omega + \int_{\partial C_i} \rho \vec{\mathbf{u}} \cdot \vec{\mathbf{n}} \mathrm{d}\Gamma &= 0 \\
\int_{C_i} \frac{\partial \rho \vec{\mathbf{u}}}{\partial t} \mathrm{d}\Omega + \int_{\partial C_i} \rho \vec{\mathbf{u}} \otimes \vec{\mathbf{u}} \cdot \vec{\mathbf{n}} \mathrm{d}\Gamma + \int_{\partial C_i} p \vec{n} \mathrm{d}\Gamma & \\
+ \frac{1}{Re} \int_{\Omega} \sigma \nabla \Phi_i \mathrm{d}\Omega + \frac{1}{Re} \int_{\Omega} \tau^{'} \nabla \Phi_i^{'} \mathrm{d}\Omega &= 0 \\
\int_{C_i} \frac{\partial E}{\partial t} \mathrm{d}\Omega + \int_{\partial C_i} (E + p) \vec{\mathbf{u}} \cdot \vec{\mathbf{n}} \mathrm{d}\Gamma + \int_{\Omega} \sigma \vec{\mathbf{u}} \cdot \nabla \Phi_i \mathrm{d}\Omega & \\
+ \frac{\gamma}{RePr} \int_{\Omega} \nabla e \cdot \nabla \Phi_i \mathrm{d}\Omega + \frac{\gamma}{RePr_t} \int_{\Omega} \mu_t^{'} \nabla e^{'} \cdot \nabla \Phi_i^{'} \mathrm{d}\Omega &= 0
\end{aligned}
\tag{3.10}
$$

where e denotes the internal energy ($E = e + \frac{1}{2}\vec{V}^2$) and $\tau^{'}$ is the small resolved scales SGS stress given by:

$$
\tau^{'} = \mu_t^{'} + (2S_{ij}^{'} - \frac{2}{3} S_{kk}^{'} \delta_{ij})
$$

with $S_{ij}^{'} = \frac{1}{2}(\frac{\partial u_i^{'}}{\partial x_j} + \frac{\partial u_j^{'}}{\partial x_i})$ and $\mu_t^{'}$ is the small resolved scales eddy viscosity (which depends on the chosen SGS model).

One can notice that the laminar Navier-Stokes equations are recovered by substituting $\tau^{'}$ by 0 and $\mu_t^{'}$ by 0 in the above Equation (3.10) and that the SGS model is recovered by substituting $\tau^{'}$ by τ, $\mu_t^{'}$ by μ_t, $e^{'}$ by e and $\Phi_i^{'}$ by Φ_i, where τ and μ_t denote the usual SGS stress tensor and SGS eddy viscosity, respectively.

More details about this VMS-LES methodology can be found in [Koobus 2004] and [C. Farhat 2005].

It remains to model $\mu_t^{'}$. We resume in the next subsection some classical SGS models which can be used for this.

3.2.3 SGS models

Smagorinsky

To approximate the SGS Reynolds stress, a SGS model can be employed. The most commonly used SGS models in LES is the Smagorinsky model in which the eddy viscosity is defined by

$$\mu_{\text{sgs}} = \overline{\rho}\,(C_{\text{s}}\Delta)^2 \left|\widetilde{S}\right| \quad , \tag{3.11}$$

where Δ is the filter width, C_{s} is the Smagorinsky coefficient and $\left|\widetilde{S}\right| = \sqrt{2\widetilde{S_{ij}}widetildeS_{ij}}$. A typical value for the Smagorinsky coefficient is $C_s = 0.1$ that is often used, especially in the presence of the mean shear.

The Smagorinsky model [Smagorinsky 1963] is certainly the simplest and most commonly used eddy-viscosity model. The major merits of the Smagorinsky model are its manageability, its computational stability and the simplicity of its formulation (involving only one adjusted parameter). All this makes it a very valuable tool for engineering applications. However, while this model is found to give acceptable results in LES of homogeneous and isotropic turbulence, with $C_s \approx 0.17$ according to Lilly [Lilly 1992], it is too dissipative with respect to the resolved motions in the near-wall region due to an excessive eddy-viscosity arising from the mean shear. The eddy-viscosity predicted by Smagorinsky model is non-zero in laminar flow regions; the model introduces spurious dissipation which damps the growth of small perturbations and thus restrains the transition to turbulence. To alleviate these deficiencies in the case of wall-bounded flows, the Smagorinsky constant C_s is often multiplied by a damping factor depending on the wall-normal distance, but no theoretical basis has been adequately adressed yet.

WALE

The second SGS model we considered is the Wall-Adapting Local Eddy-Viscosity (WALE) SGS model proposed by Nicoud and Ducros [Nicoud 1999]. The eddy-viscosity term μ_{sgs} of the model is defined by:

$$\mu_{\text{sgs}} = \overline{\rho}(C_w\Delta)^2 \frac{(\widetilde{S_{ij}}^d\widetilde{S_{ij}}^d)^{\frac{3}{2}}}{(\widetilde{S_{ij}}\widetilde{S_{ij}})^{\frac{5}{2}} + (\widetilde{S_{ij}}^d\widetilde{S_{ij}}^d)^{\frac{5}{4}}} \tag{3.12}$$

with $\widetilde{S_{ij}}^d$ being the symmetric part of the tensor ${g_{ij}}^2 = g_{ik}g_{kj}$, where $g_{ij} = \partial\tilde{u}_i/\partial x_j$:

$$\widetilde{S_{ij}}^d = \frac{1}{2}({g_{ij}}^2 + {g_{ji}}^2) - \frac{1}{3}\delta_{ij}{g_{kk}}^2.$$

For this SGS model, compared to the classical Smagorinsky formulation, the eddy-viscosity goes naturally to zero in the vicinity of the wall with the correct asymptotic behaviour so that neither (dynamic) constant adjustment nor damping function are needed to compute wall bounded flows. WALE SGS model is also capable to correctly reproduce the laminar/turbulent transition. Nevertheless, WALE model tends to add too much turbulent viscosity in vortex regions and thus too much dissipation for these regions.

Dynamic model Historically, the constants (C_s ,C_w) appearing in the expression of the viscosity of a SGS model were often arbitrarily set to a constant over the entire flow field. For general inhomogeneous flows, it can be a strongly varying functions of space. In dynamic models, these constants are replaced according to Germano *et al.* [M. Germano 1991], with a dimensionless parameter $C(\mathbf{x}, t)$ that is allowed to be a function of space and time. $C(\mathbf{x}, t)$ is estimated using information from the resolved scales making the model self-tuning. The so-called dynamic model [M. Germano 1991] has been refined [S. Ghosal 1995, Lilly 1992] over the past several years and has been successfully used to study a variety of complex inhomogeneous flows. The first step in the dynamic model consists in the introduction of a second filter, larger than the filter width, which is called the *test-filter*. The test-filter is applied to the grid filtered Navier-Stokes equations so that the *subtest-scale stress* is defined as:

$$M_{ij}^{test} = \widehat{\overline{\rho u_i u_j}} - \left(\hat{\bar{\rho}}\right)^{-1} \left(\widehat{\overline{\rho u_i}}\ \widehat{\overline{\rho u_i}}\right) \tag{3.13}$$

and, using a Smagorinsky or WALE model, it can be written:

$$M_{ij}^{test} - \frac{1}{3} M_{kk}^{test} \delta_{ij} \;=\; - C \hat{\Delta}^2 \hat{\bar{\rho}} g(\hat{\tilde{u}}) \hat{\tilde{P}}_{ij} \qquad (C \text{ denotes } {C_w}^2 \ or\ {C_s}^2) \tag{3.14}$$

We recall that the over-line denotes the grid filter, the tilde denotes the *Favre filter* and by hat we denote the test-filter. The constant C, as originally proposed by Germano *et al.* [M. Germano 1991], should be chosen such that the subgrid-scale is consistent with the subtest-scale. The quantity:

$$\mathcal{L}_{ij} = M_{ij}^{test} - \hat{M}_{ij} = \widehat{\bar{\rho} \tilde{u}_i \tilde{u}_j} - \left(\hat{\bar{\rho}}\right)^{-1} \left(\widehat{\bar{\rho} \tilde{u}_i} \widehat{\bar{\rho} \tilde{u}_i}\right) \tag{3.15}$$

is known in LES simulations. In order to determine the constant, one can compare L_{ij} to the value that would be obtained using the SGS model (Smagorinsky or WALE). This leads to

$$L_{ij} = \mathcal{L}_{ij} - \frac{1}{3} \mathcal{L}_{kk} \delta_{ij} \;=\; (C\Delta^2) B_{ij} \tag{3.16}$$

where

$$B_{ij} \;=\; \widehat{\bar{\rho} g(\tilde{u}) \tilde{P}_{ij}} - \left(\frac{\hat{\Delta}}{\Delta}\right)^2 \hat{\bar{\rho}} g(\hat{\tilde{u}}) \hat{\tilde{P}}_{ij}.$$

Equation (3.16) represents six equations with one unknown. The unknown $(C\Delta^2)$ has to satisfy:

$$L_{ij} = (C\Delta^2) B_{ij}. \tag{3.17}$$

This system of six equations can be contracted by using the least squares approach. $(C\Delta^2)$ minimizes the quantity

$$Q = (L_{ij} - (C\Delta^2)B_{ij})^2. \tag{3.18}$$

Thus, $(C\Delta^2)$ is found by setting $\frac{\partial Q}{\partial(C\Delta^2)} = 0$, from which we derive the value of $(C\Delta^2)$:

$$(C\Delta^2) = \frac{L_{ij}B_{ij}}{B_{pq}B_{pq}} \ . \tag{3.19}$$

An obstacle to the development of a dynamic version based on the Germano-identity [M. Germano 1991] could be caused by the sensitivity of SGS models having the correct behavior near the wall to the filtering procedure and the stabilization method. A simple way to avoid this inconvenient is to have a sensor able to detect the presence of the wall, without *a priori* knowledge of the geometry, so that the considered SGS model of WALE adapts the classical constant of the model, which is equal to 0.5 in the near wall region, and computes the constant dynamically otherwise. For our numerical experiments we adopt the sensor proposed by Toda and Truffin in [H. Baya Toda 2010], the expression of which is:

$$SVS = \frac{{S^d_{ij}S^d_{ij}}^{\frac{3}{2}}}{{S^d_{ij}S^d_{ij}}^{\frac{3}{2}} + {S_{ij}S_{ij}}^{3}}. \tag{3.20}$$

This parameter has the property to behave like y^{+3} (to be equal to 0 for pure shear flows and 1 for pure rotating flows).

In order to avoid high variations and important negative values of the dynamical constant (phenomena that could lead to instabilities in the simulation), we smoothen it by an averaging process over neighbour cells and a clipping procedure is applied when the sum of the SGS and the molecular viscosity is negative.

3.3 Hybrid Models

3.3.1 Model features and description

A major limitation of LES for the simulation of complex flows is the fact that its cost increases as the flow Reynolds number rises. Indeed, the grid has to be fine enough to resolve a significant part of the turbulent scales, and this becomes particularly critical in the near-wall regions where small vortical structures play a key role. One way to overcome this limitation is to introduce RANS modeling in these regions.

For this purpose, hybrid models have recently been proposed in the literature (see for example [Travin 1999, P.R. Spalart 1997, Sagaut 2002, Frohlich 2008, Vengadesan 2007, Camarri 2005]) in which RANS and LES approaches are combined together in order to obtain simulations as accurate as in the LES case but at reasonable computational cost. These hybrid methods can be divided in *zonal* approaches, in which RANS and LES are used in a-priori defined regions, and the so called *universal* models, which should be able to automatically switch from RANS to LES throughout the computational domain. In the perspective of the simulation of massively separated unsteady flows for complex geometry, as occur in many cases of engineering or industrial interest, we are primarily interested in universal hybrid models. In this context, we propose a new strategy for blending RANS and LES approaches in a hybrid model.

We begin this section by recalling some different hybrid strategies that are representative examples of universal RANS/LES hybridization before presenting the new hybrid model that we propose.

3.3.2 Detached Eddy Simulation (DES)

The original DES model proposed in [P.R. Spalart 1997] is an extension of the Spalart-Allmaras (**S-A**) RANS model, a one equation model for the turbulent viscosity ν_t:

$$\nu_t = \tilde{\nu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad \chi := \frac{\tilde{\nu}}{\nu}. \tag{3.21}$$

The S-A RANS model solves a transport equation for a viscosity-like variable $\tilde{\nu}$ often refered to as Spalart-Allmaras variable:

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} \quad = \quad & C_{b1}[1 - f_{t2}]\tilde{S}\tilde{\nu} + \frac{1}{\sigma}\{\nabla \cdot [(\nu + \tilde{\nu})\nabla\tilde{\nu}] + C_{b2}|\nabla\nu|^2\} - \qquad (3.22) \\ & \left[C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2}\right] \left(\frac{\tilde{\nu}}{d}\right)^2 + f_{t1}\Delta U^2, \end{aligned}$$

where:

$$\begin{aligned} &\tilde{S} \equiv S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \\ &S \equiv \sqrt{2\Omega_{ij}\Omega_{ij}}, \quad \Omega_{ij} \equiv \frac{1}{2}(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}), \\ &f_w = g\left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6}\right]^{1/6}, \quad g = r + C_{w2}(r^6 - r), \; r \equiv \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, \\ &f_{t1} = C_{t1} g_t \exp\left(-C_{t2}\frac{\omega_t^2}{\Delta U^2}[d^2 + g_t^2 d_t^2]\right), \quad f_{t2} = C_{t3}\exp(-C_{t4}\chi^2). \end{aligned}$$

Symbols $\sigma, C_{b1}, C_{b2}, \kappa, C_{w1}, C_{w2}, C_{w3}, C_{v1}, C_{t1}, C_{t2}, C_{t3}, C_{t4}$ hold for constants and the distance to the closest surface is denoted d.

As explained in [P.R. Spalart 1997], DES works as a sub-grid-scale model in regions where the grid density is fine enough for a LES, and as a Reynolds-averaged model in regions where it is not.

The idea is to keep the S-A model near the wall and to transform the S-A model into a one equation LES model in the remainder of the computational domain.

To accomplish this, a modified distance function is introduced as following. The standard S-A model uses the distance to the closest wall, d, as a length scale. The DES modification consists in substituting for d, everywhere in the equations, the new length scale $\tilde{d}$ defined as:

$$\tilde{d} = min \left[d, C_{\text{DES}} \Delta \right], \tag{3.23}$$

where C_{DES} is a constant usually set to 0.65 and Δ is the largest dimension of the grid cell in question.

It has become more and more clear that the main drawback of DES is its sensitivity regarding the grid. DES is well understood in thin boundary layers, with flattened grid cells and RANS mode, and in regions of massive separation, with the turbulence model in its LES mode and grid cells close to isotropic. However, DES can exhibit an incorrect behavior in thick boundary layers and shallow separation regions. This has been observed when the grid spacing parallel to the wall becomes less than the boundary layer thickness, either through grid refinement or through boundary layer thickening. In this situation, according to DES filter, LES model can be choosen, even if a RANS model is much more preferable. This could introduce also a premature separation.

In this situation, it is preferable to over-ride the DES limiter and maintain RANS behaviour in the boundary layer, independent of the grid spacing parallel to the wall relative to the thickness of the boundary layer. For this purpose a new version of this technique refered to as Delayed Detached Eddy Simulation (DDES) has been developed by Spalart and co-workers in [P.R. Spalart 2006]. A summary of its main features is presented next.

3.3.3 Delayed-Detached Eddy Simulation (DDES)

The purpose of DDES, as explained earlier, is to "delay" LES function or to "preserve" RANS mode in boundary layer.
To achieve this, Spalart *et al.* proposed in [P.R. Spalart 2006] to use a blending function having as argument a parameter ratio of a RANS model length scale to the wall distance. This parameter for DDES is defined as:

$$r_d \equiv \frac{\nu_t + \nu}{\sqrt{u_{i,j} u_{i,j}} \kappa^2 d^2}, \tag{3.24}$$

where ν_t is the kinematic eddy viscosity, ν the kinetic molecular viscosity, $u_{i,j}$ the velocity gradients, κ the Kármán constant, and d the distance to the wall.
The function of blending is defined then as:

$$f_d \equiv 1 - tanh([8r_d]^3), \tag{3.25}$$

which is designed to be 1 in the LES region where $r_d << 1$, and 0 elsewhere (and to be insensitive to r_d exceeding 1 very near the wall).

The application of this procedure to S-A-based DES, means re-defining the DES length scale $\tilde{d}$:

$$\tilde{d} \equiv d - f_d max(0, d - C_{DES}\Delta) \tag{3.26}$$

Setting f_d to zero yields RANS, while setting it to 1 gives DES ($\tilde{d} = min(d, C_{DES}\Delta)$).

3.3.4 Limited Numerical Scales (LNS) Approach

The basic idea of the LNS model [P. Batten 2000] is to multiply the Reynolds stress tensor, given by the RANS closure, by a blending function, which allows to switch from the RANS to the LES approach.

Validation of this hybrid approach in a finite volume/finite element framework on unstructured meshes can be found in [Camarri 2005, Koobus 2007]. In these works, the $k-\varepsilon$ model [Launder 1979] is used for the RANS closure, in which the Reynolds stress tensor is modeled as follows, by introducing a turbulent eddy-viscosity, μ_t:

$$R_{ij} \simeq \mu_{\mathrm{t}} \underbrace{\left[\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i} - \frac{2}{3}\frac{\partial \widetilde{u}_l}{\partial x_l}\delta_{ij}\right]}_{\widetilde{P}_{ij}} - \frac{2}{3}\langle\rho\rangle k \delta_{ij} \quad , \tag{3.27}$$

where the tilde denotes the Favre average, the $< \cdot >$ time averaging, δ_{ij} is the Krönecker symbol and k is the turbulent kinetic energy. The turbulent eddy-viscosity μ_{t} is defined as a function of k and of the turbulent dissipation rate of energy, ε, as follows:

$$\mu_{\mathrm{t}} = C_\mu \frac{k^2}{\varepsilon} \quad , \tag{3.28}$$

in which C_μ is a model parameter, set equal to the classical value of 0.09 and k and ε are obtained from the corresponding modeled transport equations [Launder 1979].

The LNS equations are then obtained from the RANS equations by replacing the Reynolds stress tensor R_{ij}, given by Equation (3.27), with the tensor L_{ij}:

$$L_{ij} = \alpha R_{ij} = \alpha \mu_t \widetilde{P}_{ij} - \frac{2}{3}\langle\rho\rangle \left(\alpha k\right) \delta_{ij} \quad , \tag{3.29}$$

where α is the damping function ($0 \leq \alpha \leq 1$), varying in space and time.

In the LNS model, the damping function is defined as follows:

$$\alpha = \min\left\{\frac{\mu_{sgs}}{\mu_t}, 1\right\} \tag{3.30}$$

in which μ_{sgs} is the SGS viscosity obtained from a LES closure model.

As discussed in [P. Batten 2004], the model should work in the LES mode where the grid is fine enough to resolve a significant part of the turbulence scales, as in LES; elsewhere ($\alpha = 1$), the $k - \varepsilon$ RANS closure should be recovered.

3.4 Our hybrid RANS/VMS-LES model

In this section our strategy for blending RANS and VMS-LES approach into a new hybrid model is presented. This strategy of hybridisation permits a natural integration of the VMS concept, while this is not yet the case for other existing approaches, like LNS and DNS.

To start with, as explained in [Sagaut 2002], the flow variables are decomposed in a RANS part (*i.e.* the averaged flow field), a correction part that takes into account the turbulent large-scale fluctuations, and a third part made of the unresolved or SGS fluctuations. The basic idea is to solve the RANS equations in the whole computational domain and to correct the obtained averaged flow field by adding, where the grid is adequately refined, the remaining resolved fluctuations. We search here for a hybridization strategy in which the RANS and LES models are blended in the computational domain following a given criterion. To this aim, a blending function is introduced, θ, which smoothly varies between 0 and 1. The correction term which is added to the averaged flow field is thus damped by a factor $1-\theta$, obtaining a model which coincides with the RANS approach when $\theta = 1$ and recovers the LES aproach in the limit of $\theta \to 0$. Following strictly these guidelines would imply that the two fields, RANS and LES correction, need to be computed separately. In this thesis we explore a single field version. We start with describing the two main components, the RANS and LES models, and then we specify how we define our blending field θ.

3.4.1 Hybrid RANS/LES Coupling

The Navier-Stokes equations for compressible flows of perfect Newtonian gases are considered here, in conservative form and using the following variables $W = (\rho, \rho u_i, E)$ where ρ is the density, ρu_i, $i = 1, 2, 3$ are momentum variables and $E = \rho e + 1/2\rho u_i u_i$ is the total energy per unit volume, e being the internal energy. As in [Sagaut 2002], the following decomposition of the flow variables is adopted:

$$W = \underbrace{< W >}_{RANS} + \underbrace{W^c}_{correction} + W^{SGS}$$

where $< W >$ are the RANS flow variables, obtained by applying an averaging operator to the Navier-Stokes equations, W^c are the remaining resolved fluctuations (*i.e.* $< W > + W^c$ are the flow variables in LES) and W^{SGS} are the unresolved or SGS fluctuations.

If we write the Navier-Stokes equations in the following compact conservative form:

$$\frac{\partial W}{\partial t} + \nabla \cdot \mathcal{F}(W) = 0$$

in which $\mathcal{F}$ represents both the viscous and the convective fluxes, for the averaged flow $< W >$ we get:

$$\frac{\partial < W >}{\partial t} + \nabla \cdot \mathcal{F}(< W >) = -\tau^{RANS}(< W >) \tag{3.31}$$

where $\tau^{RANS}(< W >)$ is the closure term given by a RANS turbulence model.

As it was previously explained, by applying a filtering operator to the Navier-Stokes equations, the LES equations are obtained, which can be written as follows:

$$\frac{\partial(< W > + W^c)}{\partial t} + \nabla \cdot \mathcal{F}(< W > + W^c) = -\tau^{LES}(< W > + W^c) \tag{3.32}$$

where τ^{LES} is the SGS term.

An equation for the resolved fluctuations W^c can thus be derived (see also [Sagaut 2002]):

$$\begin{aligned} &\frac{\partial W^c}{\partial t} + \nabla \cdot \mathcal{F}(< W > + W^c) - \nabla \cdot \mathcal{F}(< W >) \\ &\quad = \tau^{RANS}(< W >) - \tau^{LES}(< W > + W^c). \end{aligned} \tag{3.33}$$

The basic idea of the proposed hybrid model is to solve System (3.31) in the whole domain and to correct the obtained averaged flow by adding the remaining resolved fluctuations (computed through Equation (3.33)), wherever the grid resolution is adequate for LES. To identify the regions where the additional fluctuations must be computed, we introduce a blending function θ smoothly varying between 0 and 1. When $\theta = 1$, no correction to $< W >$ is computed and, thus, the RANS approach is recovered. Conversely, wherever $\theta < 1$, additional resolved fluctuations are computed; in the limit of $\theta \to 0$ we want to recover a full LES approach. Thus, the following equation is used for the correction term:

$$\begin{aligned} &\frac{\partial W^c}{\partial t} + \nabla \cdot \mathcal{F}(< W > + W^c) - \nabla \cdot \mathcal{F}(< W >) \\ &= (1-\theta)\left[\tau^{RANS}(< W >) - \tau^{LES}(< W > + W^c)\right]. \end{aligned} \tag{3.34}$$

Note that for $\theta \to 0$ the RANS limit is actually recovered; indeed, for $\theta = 1$ the right-hand side of System (3.34) vanishes and, hence, a trivial solution is $W^c = 0$. As required, for $\theta = 0$ System (3.34) becomes identical to System (3.33) and the remaining resolved fluctuations are added to the averaged flow; the model, thus, works in LES mode. For θ going from 1 to 0, *i.e.* when, following the definition of the blending function (cf. Relation 3.55), the grid resolution is intermediate between one adequate for RANS and one adequate for LES, the right-hand side term in Equation (3.34) is damped through multiplication by $(1 - \theta)$. Although, this seems rather arbitrary from a physical point of view, this is aimed to obtain a smooth transition between RANS and LES. More specifically, we wish to obtain a progressive addition of fluctuations when the grid resolution increases and the model switches from the RANS to the LES mode. Summarizing, the ingredients of the proposed approach are: a RANS closure model, a SGS model for LES and the definition of the blending function. These will be described in Sections 3.4.2 and 3.5.

3.4.2 RANS component

Concerning the RANS closure model used in the hybrid model, the $k - \varepsilon$ model [Launder 1979] is chosen. The main ingredients of this closure model are presented next.

Standard $k - \varepsilon$ model. This is an eddy viscosity model with a turbulent viscosity μ_t defined from two extra variables (two-equation model) the turbulent kinetic energy k and its dissipation rate ε, and given by expression (3.28). The Reynolds tensor, the main term to model, is then given by Equation (3.27).

The k and ε fields are evaluated by solving the two extra transport equations:

$$\frac{\partial \langle\rho\rangle k}{\partial t} + \left(\langle\rho\rangle \widetilde{u}_j k\right)_{,j} = \left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]_{,j} + R_{ij}\frac{\partial \widetilde{u}_i}{\partial x_j} - \langle\rho\rangle\varepsilon \quad , \tag{3.35}$$

$$\frac{\partial \langle\rho\rangle \varepsilon}{\partial t} + \left(\langle\rho\rangle \varepsilon \widetilde{u}_j\right)_{,j} = \left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right]_{,j} + C_{\varepsilon 1}\left(\frac{\varepsilon}{k}\right) R_{ij}\frac{\partial \widetilde{u}_i}{\partial x_j} - C_{\varepsilon 2}\langle\rho\rangle\frac{\varepsilon^2}{k} \quad . \tag{3.36}$$

where constants $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, σ_k and σ_ε are defined by:

$$C_{\varepsilon 1} = 1.44 \qquad C_{\varepsilon 2} = 1.92 \qquad \sigma_k = 1.0 \qquad \sigma_\varepsilon = 1.3$$

Historically these constants are deduced from the application of the model to simple turbulent flows. They also can be mathematically derived by the Renormalization Group method, see [Yakhot 1986].

Wall treatment by wall law. The k and ε variables, the velocity and the temperature can have a stiff behavior and exhibit small scales near a wall. However, their behavior presents in most cases some common features, such as obeying the logarithmic law for the tangent velocity component. In the theory and in the $k - \varepsilon$ formulation, this is a consequence of the equilibrium between turbulent kinetic energy production and dissipation, which arises in a large enough region close to wall. Wall law methods use this feature in order to avoid the costly discretization of the wall behavior of these variables. Closer to the wall, two different behaviors need to be taken into account: in the sublayer and in the *log* region. This is well modeled in Reichardt's law (see for example [Hinze 1959]) which writes in terms of U^+ and y^+:

$$U^+ = \frac{\overline{U}}{U_\tau} \quad , \tag{3.37}$$

$$y^+ = \frac{\rho U_\tau}{\mu} y \quad . \tag{3.38}$$

where $\overline{U}$ is the statistical average of the velocity and the friction velocity U_τ is given by:

$$U_\tau = \sqrt{\frac{\tau_p}{\rho}} \quad . \tag{3.39}$$

Then we set:

$$U^+ = \frac{1}{\kappa} ln\left(1 + \kappa y^+\right) + 7.8\left(1 - e^{-\frac{y^+}{11}} - \frac{y^+}{11} e^{-0.33 y^+}\right) \quad . \tag{3.40}$$

where $\kappa = 0.41$ is the von Kàrmàn constant.

In practice, the computational domain is restricted to a domain at a distance d to the wall, with $d > \delta$, where δ is prescribed by the user. The part $d < \delta$ of the flow is represented by the law (3.40) after evaluation of U_τ and matched to the numerical solution in the computational domain. The matching distance δ is fixed by the user in such a way that its normalised counterpart $y^+_{match} = \delta \cdot u_f / \gamma$ is in the log region, more precisely:

$$30 < y^+_{match} < 200. \tag{3.41}$$

An approximative value of δ is chosen by an empirical formula:

$$\delta = 20 \cdot D \cdot y^+_{match} / Rey \tag{3.42}$$

where y^+_{match} is chosen by Relation 3.41. In practice the y^+ of the matching zone varies more or less in the interval $]\,10 - 300\,[$. D is a characteristic of the dimension of the flow. In our examples, D is the diameter of the obstacle.

A Low Reynolds version. While the wall law is efficient from the CPU standpoint, better prediction close to the wall, and in particular near the separation, is obtained by resolving down to the wall, *i.e.*, applying a low Reynolds closure.
The low Reynolds $k - \varepsilon$ model proposed in [Goldberg 1998][Goldberg 1990] is presented now. Let:

$$\mu_t = C_\mu f_\mu \rho \frac{k^2}{\varepsilon} \tag{3.43}$$

where the coefficient $C_\mu = 0.09$ and f_μ is a so-called damping function (for damping the effect of model when closer to wall) defined by:

$$f_\mu = \frac{1 - e^{-A_\mu R_t}}{1 - e^{-R_t^{1/2}}} max(1, \psi^{-1}) \tag{3.44}$$

with the constant $A_\mu = 0.01$ and $\psi = R_t^{1/2} / C_\tau$. The turbulent Reynolds number :

$$R_t = k^2/(\nu\varepsilon) \;\; \text{with} \;\; \nu = \mu/\rho \tag{3.45}$$

allows us to avoid the estimation of the distance to the wall.
Then, k and ε are solutions of:

$$\frac{\partial\langle\rho\rangle k}{\partial t} + (\langle\rho\rangle\widetilde{u}_j k)_{,j} = \left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]_{,j} + R_{ij}\frac{\partial\widetilde{u}_i}{\partial x_j} - \langle\rho\rangle\varepsilon \quad , \tag{3.46}$$

$$\frac{\partial\langle\rho\rangle\varepsilon}{\partial t} + (\langle\rho\rangle\varepsilon\widetilde{u}_j)_{,j} = \left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial\varepsilon}{\partial x_j}\right]_{,j} + \left(C_{\varepsilon 1}R_{ij}\frac{\partial\widetilde{u}_i}{\partial x_j} - C_{\varepsilon 2}\langle\rho\rangle\varepsilon + E\right)T_\tau^{-1} \quad . \tag{3.47}$$

where T_τ is the *realizable time scale*:

$$T_\tau = \frac{k}{\varepsilon}max(1, \psi^{-1}). \tag{3.48}$$

This time scale is k/ε for large values of turbulent Reynolds number R_t and thus of ψ, but tends to be equal to the Kolmogorov scale $C_\tau(\nu/\varepsilon)^{1/2}$ for $R_t << 1$. Constants are defined as $C_\tau = 1.41$, $C_{\varepsilon 1} = 1.42$, $C_{\varepsilon 2} = 1.83$ and:

$$E = \rho A_E V(\varepsilon T_\tau)^{0.5}\xi \tag{3.49}$$

with $A_E = 0.3$, $V = max(\sqrt{k},\ (\nu\varepsilon)^{0.25})$ and $\xi = max(\frac{\partial k}{\partial x_i}\frac{\partial\tau}{\partial x_i},\ 0)$, where $\tau = k/\varepsilon$.

One positive point of this precise low-Reynolds model is that it does not require a distance to the wall, which can be cumbersome to evaluate in the case of unstructured meshes and complex geometries are those encountred in industrial applications.
In case of very high Reynolds number demanding a very stretched mesh and multiples layers on a small thickness around the obstacle, the discretization matrix for low-Reynolds models is in general badly conditioned and the system difficult to solve.

In the sequel we combine this low Reynolds model to the wall law. This is useful because in practice, the approximative empiric choice of the layer length δ does not ensures that small y^+ (smaller that 30) cannot be encountered in some part of the flow. In the regions where y^+ is small, the Goldberg model applies and improves the flow prediction with respect to the regular $k - \varepsilon$ model.

3.5 LES component : VMS-LES

Regarding the LES mode, we wish to recover the Variational Multi-Scale approach [T.J.R. Hughes 2000], in which the flow variables are decomposed as follows:

$$W = \underbrace{\bar{W}}_{LRS} + \underbrace{W'}_{SRS} + W^{SGS} \tag{3.50}$$

where $\bar{W}$ are the large resolved scales (LRS) and W' are the small resolved scales (SRS). This decomposition is obtained by variational projection in the LRS and SRS spaces respectively. In the present study, we follow the VMS approach as in Section 3.2. Let ψ_l and ϕ_l be the finite-volume and respectively finite-element basis functions associated to the used grid. Let $\psi_l^{'}$ and $\phi_l^{'}$ be the finite volume and finite element basis functions associated to the largest and smallest resolved scales. Thus, the VMS-LES formulation writes:

$$\begin{aligned}\left(\frac{\partial(< W > +W^c)}{\partial t}, \psi_l\right) + (\nabla \cdot F_c(< W > +W^c), \psi_l) \\ + (\nabla \cdot F_v(< W > +W^c), \phi_l) = -\left(\tau^{LES}(W'), \phi_l^{'}\right) \quad l = 1, N\end{aligned} \tag{3.51}$$

in which F_c holds for the convective fluxes, F_v for diffusive fluxes and τ^{LES} is modeled by introducing the Smagorinsky SGS eddy-viscosity μ_s:

$$\mu_s \; = \; \rho' C_s \Delta^2 \sqrt{S_{ij}^{'} \cdot S_{ij}^{'}} \tag{3.52}$$

where C_s is the model input parameter, $S_{ij}^{'}$ is the strain-rate tensor (computed in the VMS approach as a function of W') and Δ is a length which should be representative of the size of the resolved turbulent scales. Here, Δ has been selected as $Vol(i)^{1/3}$ ($Vol(i)$ being the volume of the i-th tetrahedron) and C_s has been set equal to 0.1.

3.5.1 Hybrid Complete Formulation

Finally, the Galerkin projection of Equations (3.31) and (3.51) for the computation of $< W >$ and of the additional fluctuations in the proposed hybrid model become respectively:

$$\begin{aligned}\left(\frac{\partial < W >}{\partial t}, \psi_l\right) + (\nabla \cdot F_c(< W >), \psi_l) + (\nabla \cdot F_v(< W >), \phi_l) \\ = -\left(\tau^{RANS}(< W >), \phi_l\right) \quad l = 1, N\end{aligned} \tag{3.53}$$

$$\begin{aligned}\left(\frac{\partial W^c}{\partial t}, \psi_l\right) + (\nabla \cdot F_c(< W > +W^c).\psi_l) - (\nabla \cdot F_c(< W >), \psi_l) + (\nabla \cdot F_v(W^c), \phi_l) \\ = (1-\theta)\left[\left(\tau^{RANS}(< W >), \phi_l\right) - \left(\tau^{LES}(W'), \phi_l\right)\right] \quad l = 1, N\end{aligned} \tag{3.54}$$

As a possible choice for θ, the following function is used in the present study:

$$\theta = F(\xi) = tanh(\xi^2) \tag{3.55}$$

where ξ is the *blending parameter*, which should indicate whether the grid resolution is fine enough to resolve a significant part of the turbulence fluctuations, and therefore to obtain a LES-like simulation. The choice of the *blending parameter* is clearly a key point for the definition of the present hybrid model. In this study, the blending parameter based on the ratio between characteristic length of the two models is chosen, similar to Menter and Kuntz [Menter 2004]. The accuracy of the LES model is evidently related to the local characteristic length which is the mesh size:

$$l_{LES} = \Delta$$

with $\Delta = (Vol(T))^{\frac{1}{3}}$ (T denotes the tethraedron of the mesh). For the $k - \varepsilon$ RANS model, the length scale is defined from the variables k and ε by using a dimensional argument:

$$l_{RANS} = \frac{k^{3/2}}{\varepsilon}.$$

Then we define:

$$\xi = \frac{l_{LES}}{l_{RANS}} = \varepsilon k^{-3/2}\Delta.$$

A last term needs to be accounted for near the wall boundary.

Protection zone.
As explained previously in the discussion of DES, the boundary layer is wished to be treated with RANS model, that is why the definition of the blending function is carefully studied so that no LES computation occurs for massively separated flows when the grid resolution could be misleading. This remark applies to any hybrid model of RANS/LES type. Particulary in our model, similary to a DDES approach we define a *protection zone* by introducing a distance from the wall function denoted arg:

$$arg = \frac{10 * l}{|d| + l/1000}. \tag{3.56}$$

In the expression of arg we denoted d the distance to the body surface. l is the thickness parameter, that we defined as 5 times the thickness of the wall law or matching zone δ ($l = 5\delta$). This is designed in order to include the boundary layer into the RANS zone of the hybrid formulation. The definition of this zone needs the distance d to the wall and we hope to avoid this in the future, for example by the use of turbulent Reynolds number defined by Relation (3.45).

Then, the new blending function $\bar{\theta}$ is defined as:

$$\bar{\theta} = \max(\theta, tanh(arg)). \tag{3.57}$$

Thus, when $|d| \to 0$ in (3.56) then $\bar{\theta} \approx 1$ and when on the contrary $|d| \to \infty$ then $\bar{\theta} \approx 0$.

Single field formulation.
To avoid the solution of two different systems of PDEs and the consequent increase

of required computational resources, Systems (3.53) and (3.54) have been recast together as:

$$\begin{aligned}&\left(\frac{\partial W}{\partial t},\psi_l\right)+(\nabla\cdot F_c(W),\psi_l)+(\nabla\cdot F_v(W),\psi_l)\\&=-\theta\left(\tau^{RANS}(<W>),\phi_l\right)-(1-\theta)\left(\tau^{LES}(W'),\phi_l'\right)\quad l=1,N\end{aligned}\tag{3.58}$$

Clearly, if only Equation (3.58) is solved, $< W >$ is not available at each time step. Two different options are possible: either to use an approximation of $< W >$ obtained by averaging and smoothing of W, in the spirit of VMS, or to simply use W in Equation (3.58). This second option is used here, by relying on the tendancy of RANS to naturally damp fluctuations.

3.6 Consistent Mass Matrix study of a subcritical flow

The purpose of this section is to analyse the influence of the consistent mass matrix numerical scheme introduced in Section 2.4. We compare it with the basic scheme. This work was done in collaboration with Hilde Ouvrard.

VMS-LES is performed to simulate the flow past a circular cylinder at Mach number $M_\infty = 0.1$ and at a subcritical Reynolds number Rey of 3900 ($Rey = \frac{u_\infty D}{\nu}$) based on cylinder diameter D and free-steam velocity u_∞. This Reynolds number corresponds to the intermediate subscritical regime in which the drag is at a local minimum around $Rey = 2000 - 3000$ and increases slowly to $Rey = 20000$ (see [Achenbach 1968]). Noticeable variations of the lift fluctuations and mean base pressure are observed while Strouhal number is marginally decreasing. The main instability in the flow is Von Karman wake instability. At some Reynolds the Kelvin-Helmotz instability of the free shear-layers appears. They are generally not observed for Reynolds smaller that 5000, probably because between Rey 1000 and 5000 the Kelvin Helmholz vortical structure appears in a very intermitent way.

Many measurements are available in the interval $3900 - 4020$, see for example [Dong 2006, Ong 1996, P. Parneaudeau 2008, Norberg 1987] .

The corresponding flow at $Rey = 3900$ has been computed either by LES computation [Kravchenko 2000, P. Parneaudeau 2008, Ouvrard 2010] either by DNS [Dong 2006].

The computational domain as shown in Figure 3.3 is $-10 \leq x/D \leq 25$, $-20 \leq y/D \leq 20$ and $-\pi/2 \leq z/D \leq \pi/2$ where x, y and z denote the streamwise, transverse and spanwise direction respectively. The characteristics of the domain are the following:

$$L_i/D = 10, \quad L_0/D = 25, \quad H_y/D = 20 \quad \text{and} \quad H_z/D = \pi$$

The cylinder of unit diameter is centered on $(x, y) = (0, 0)$.

The flow domain is discretized by an unstructured tetrahedral grid which consists of approximatively 2.9×10^5 nodes. The averaged distance of the nearest point to the cylinder boundary is $0.017D$ which corresponds to $y^+ \approx 3.31$.

For these simulations, the characteristic Steger-Warming splitting [Steger 1981] is used at the inflow and outflow as well as on the upper and lower surface ($y = \pm H_y$). In the spanwise direction periodic boundary conditions are applied. On the cylinder surface no-slip boundary conditions are set.

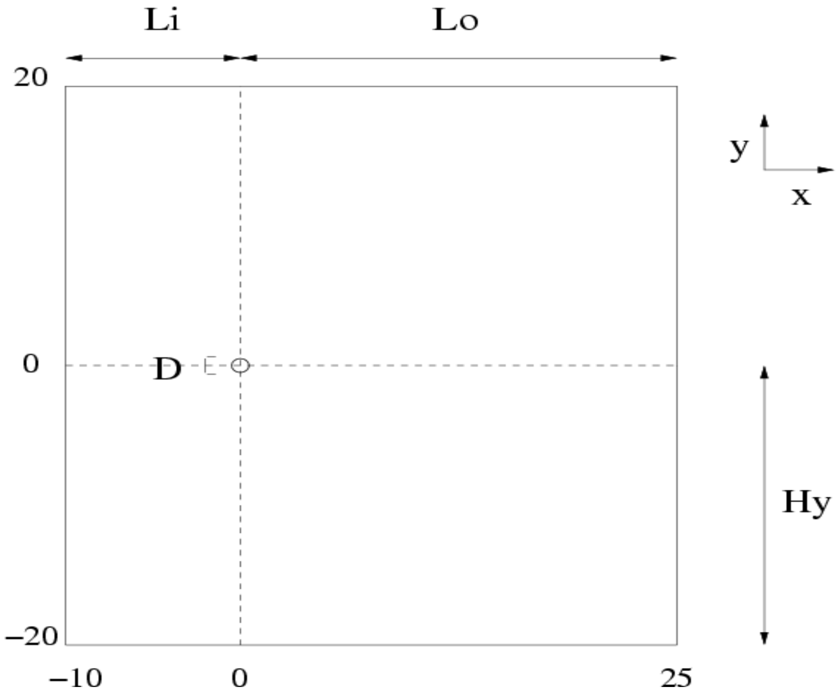


Figure 3.3: Computational domain

To investigate the influence of numerical schemes on the VMS-LES approach, three simulations have been carried out using the WALE subgrid-scale model. The characteristics of the simulations performed for this study are summarized in Table 3.1. One uses the Roe-Turkel solver as low Mach preconditioning [Guillard 1999]. For stability reasons, the mass-matrix scheme has been run setting the numerical viscosity parameter γ to its maximal value 1.

Simulation	Numerical Scheme	C	Dissipation	γ	CFL
Simu1	**Mass Matrix + Central Diff.**	$\frac{1}{30}$	**Non projected**	**1**	**20**
Simu2	**Mass Lumping + V6**	$\frac{1}{30}$	**Non projected**	**1**	**20**
Simu3	**Mass Lumping + V6**	$\frac{1}{30}$	**Non projected**	**0.3**	**20**
Simu4	**Mass Matrix + Central differencing**	$\frac{1}{15}$	**Non projected**	**1**	**20**

Table 3.1: Simulations data for VMS-LES computation with WALE subgrid-scale model, constant C defined in Section 2.4.2.

The CFL number has been chosen so that a vortex shedding cycle is sampled in a little less than 1000 time steps. Time-averaged values and turbulence parameters are summarized in Table 3.2 and compared to data from experiments of Norberg [Norberg 2003], Ong and Wallace [Ong 1996] and Parnaudeau et al.[P. Parneaudeau 2008]. $\overline{C_d}$ denotes the mean drag coefficient, C_d' and C_l' respectively the root mean square values of the drag and lift, St the Strouhal number, U_{min} the mean centerline streamwise velocity, Cp_{base} the mean base-pressure coefficient and l_r the mean recirculation length.

	St	$\overline{C_d}$	C_d'	C_l'	U_{min}	Cp_{back}	l_r
Simulations							
Simu1	**0.252**	**0.77**	**.0157**	**.0367**	**-0.20**	**-0.66**	**1.76**
Simu2	**0.213**	**1.08**	**.0490**	**.4912**	**-0.29**	**-1.09**	**0.83**
Simu3	**0.215**	**1.01**	**.0639**	**.5920**	**-0.29**	**-1.02**	**1.08**
Simu4	**0.215**	**1.02**	**.0103**	**.0303**	**-0.29**	**-0.94**	**1.12**
Experiments							
[Norberg 2003]	.215±.05	.99±.05			-.24±0.1	-.88±.05	
[Ong 1996]	.21±.005						1.4 ±0.1
[P. Parneaudeau 2008]					-.34		1.51

Table 3.2: Flow parameters for the present simulations at $Rey = 3900$

As shown in Table 3.2, for the same value of the C constant (Simu1, Simu2 and Simu3), the flow parameters obtained with the mass-matrix scheme are generally less well predicted than those obtained with the classical V6 scheme. In particular the mean drag is under-estimated with the mass-matrix scheme, which involves an over-estimation of the recirculation length. Comparing Simu1 and Simu4, one can note that introducing a twice larger numerical dissipation significantly improve the results for the mass-matrix/central-differencing approach.

Figure 3.4 shows a profile of time-averaged and z-averaged streamwise velocity. One can determine from this plot the most dissipative scheme, that is the scheme producing the shortest recirculation length. This assumption is verified on Figure 3.5, showing the pressure distribution on the cylinder surface averaged in time on homogeneous z direction. From these two plots, one can conclude that the mass-matrix scheme is not enough dissipative. This can explain the encountered robustness problem. Introducing a larger dissipation level reduces considerably the recirculation length. One can also note in Figure 3.5 the discrepancies with the experimental data, for θ being between 60 and 100. This is due to the coarseness of the used grid (see [Ouvrard 2010]).

Figure 3.6 displays the total resolved Reynolds stress $< u\prime u\prime >$ in the wake at $x = 1.54$ and Figure 3.7 displays the total resolved Reynolds stress $< v\prime v\prime >$ at $x = 1.54$. In Figure 3.6 one can observe that the plot is not symmetric using the mass-matrix scheme. We interprete this as an unsufficient balance between the

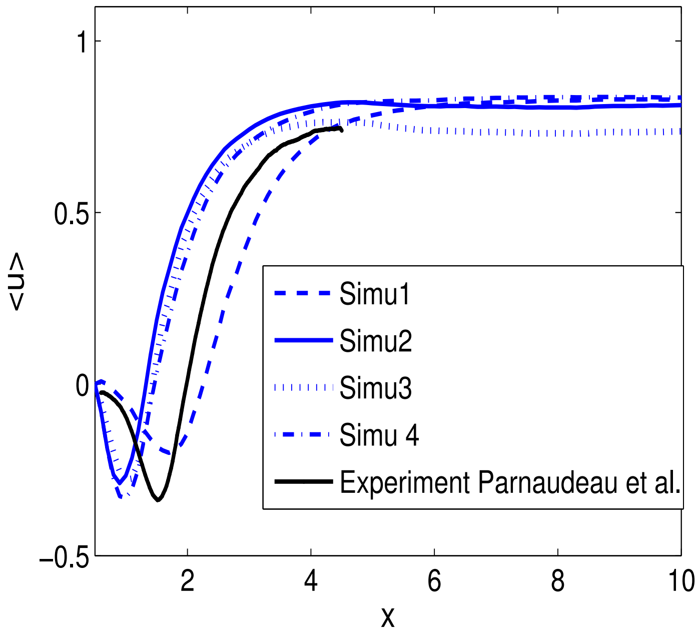


Figure 3.4: Time-averaged streamwise centerlined velocity at $Rey = 3900$

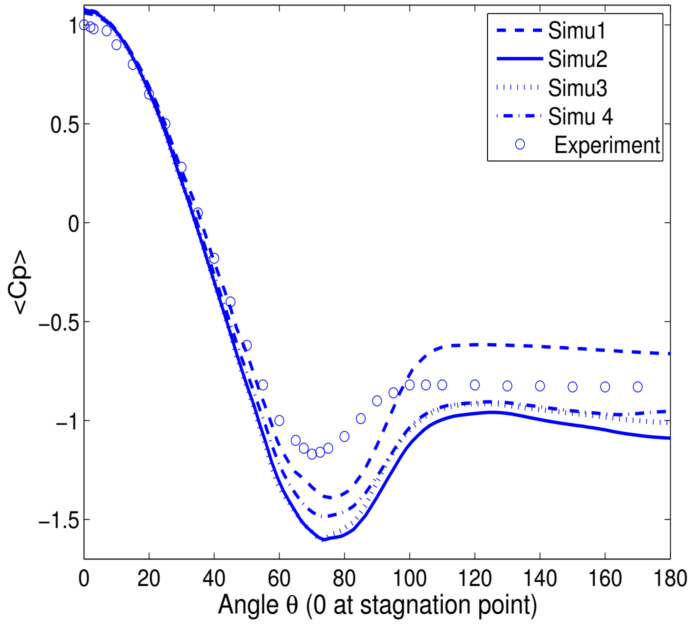


Figure 3.5: Time-averaged and z-averaged pressure distribution on the surface of the cylinder, experiment: Norberg [Norberg 2003]

compressive (anti-dissipative) effect of the constant mass matrix on one side, and the added artificial dissipation on the other side. With a low dissipation level, better results are obtained with the V6 scheme. The behaviour of the Mass-Matrix scheme is improved using a larger numerical viscosity.

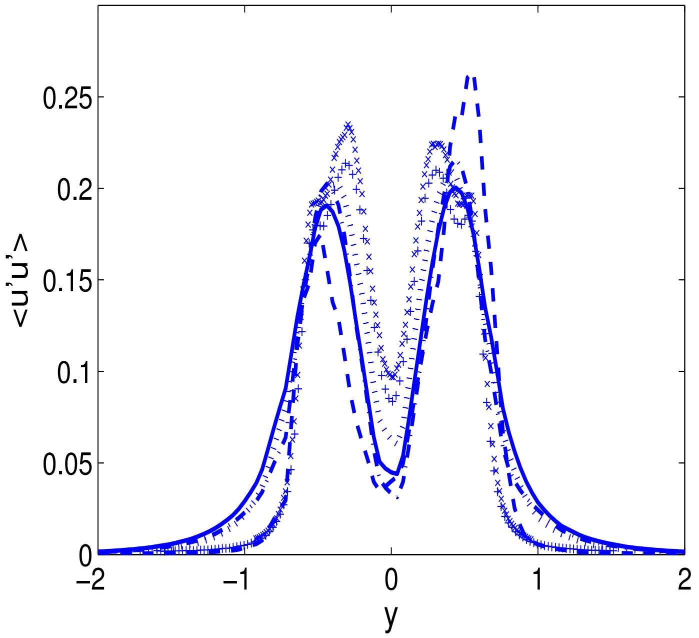


Figure 3.6: Total resolved streamwise Reynolds stress $< u\prime u\prime >$ at $x = 1.54$, experiments of Parnaudeau *et al.* [P. Parneaudeau 2008]; −−: Simu1; −: Simu2;...: Simu3; −.: Simu4; +: PIV1 experiment; ×: PIV2 experiment

In the case of total resolved streamwise Reynolds stress $< v\prime v\prime >$ at $x = 1.54$, one can notice first that for the three simulations, we are getting far away from the experiments. The mass lumping V6 scheme still yields better results than the two other ones. But increasing the numerical viscosity for the Mass-Matrix scheme (Simu4) gives a better agreement with the experiments.

3.6.1 Conclusion

The consistent mass matrix scheme represents a method where the better accuracy for very irregular meshes has been proved in Chapter 2.4. The dissipation term that has been developed here works properly for LES. However, for the considered benchmark problem and the used grid resolution and smoothness, the gain in terms of accuracy is not significant. Another study introducing ENO (Essentialy Non-Oscillatory) schemes ([Ouvrard 2009]) is now investigated and could be prefered for the improvement of our software AIRONUM.

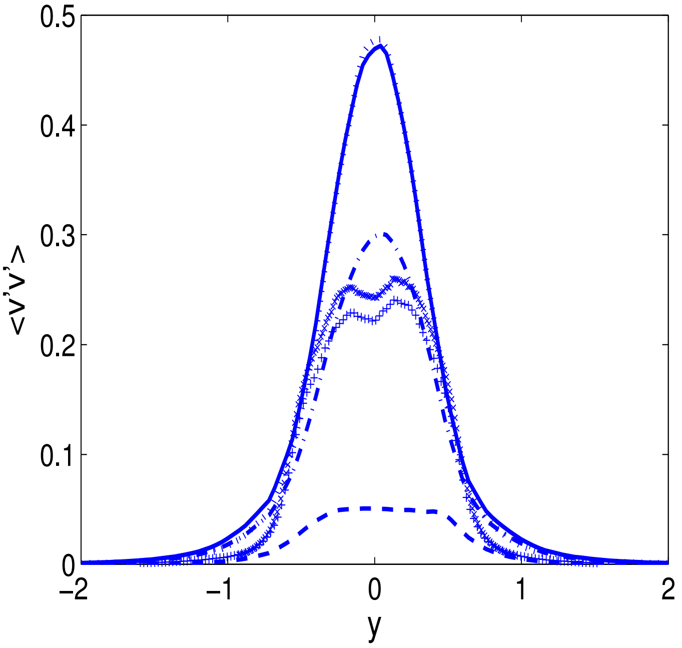


Figure 3.7: Total resolved streamwise Reynolds stress $< v\prime v\prime >$ at $x = 1.54$, experiments of Parnaudeau *et al.* [P. Parneaudeau 2008];−−: Simu1; −: Simu2;...: Simu3; −.: Simu4; +: PIV1 experiment; ×: PIV2 experiment

3.7 Hybrid computation of supercritical flow

The application of a hybrid formulation to the simulation of the flow around bluff bodies at high Reynolds numbers of the order of millions is a difficult task because of the thin boundary layer to capture. Indeed, with the Reynolds number increasing, the boundary layer becomes turbulent and the angle of separation increases. This section is devoted to a first application of our hybrid model to this class of problems. As explained in the introduction of this chapter, in order to improve the industrial efficiency for high Reynolds flow, we investigate the possibility to exploit as much as possible the wall law and to use very coarse meshes. One of the consequences in using coarse grids is that von Kármán instabilities cannot be well captured, but using fine grids is limited because of computational costs, thus a compromise has to be found.

Our benchmark test concerns the incompressible flow around a circular cylinder. For this kind of flow one observes the drag crisis, a phenomenon in which drag coefficient drops off suddenly as Reynolds number increases. This corresponds to the point where the flow pattern changes, leaving a narrower turbulent wake. This is described by Roshko in 1961 [Roshko 1961] and Achenbach in 1968 [Achenbach 1968]. The drag crisis occurs for Reynolds numbers between $200K$ and $300K$. Our computations deal with supercritical Reynolds numbers ranging between $5 \cdot 10^5$ and $3 \cdot 10^6$.

For these Reynolds numbers the flow remains sensitive to the value of Reynolds number, but only a few investigations have been realised for Reynolds number more that $5 \cdot 10^6$, although this is an interesting interval. Indeed, inside the supercritical regime the separation becomes turbulent. It is generally accepted that periodic vortex sheddings is very weak (if any) in the lowest parts of the interval. The range $0.48 \cdot 10^6$ to $7 \cdot 10^6$ has been measured by van Nunen in 1972 [van Nunen 1974]. At the lower value $0.48 \cdot 10^6$ a fairly downstream turbulent separation is observed. With an increasing Reynolds number, the final separation point moves forward. This corresponds to a slight increase of the drag.

According to Jones [Jones 1969] the interval $1.1 \cdot 10^6 < Rey < 3.5 \cdot 10^6$ is categorised into the wide-band random regime. Van Nunen suggests that a better range for this regime is $0.5 \cdot 10^6 - 2.5 \cdot 10^6$. In that regime essentially no periodicity is observed for the unsteady lift or the unsteady pressures.

The narrow-band random regime corresponds to: $2.5 \cdot 10^6 - 6.5 \cdot 10^6$. In this regime a small peak could be observed in the unsteady lift frequency analysis at Strouhal $St = 0.20$. This peak frequency is found in the region around the upstream part of the cylinder and not beyond the separation point.

Measurements of these flows can be found in [Roshko 1961, Achenbach 1968, van Nunen 1974]. Besides these authors, other measurements can be found in [Schewe 1983, Shih 1993, Warschauer 1971, Zdravkovich 1997].

These flows have been computed by a few authors. Travin *et al.* use the Detached Eddy Simulation (DES) method [Travin 1999, Wang 2001] on a mesh of about $400K$ vertices. They obtained pressures that are "consistently lower than in the majority of the experiments". Catalano *et al.* [Wang 2001] have used the combination of the LES with a wall model on a mesh of $2.3M$ nodes and could not predict good sensitivities to Reynolds numbers.

The geometrical structure of our computational domain is the same as in Figure 3.3. Periodic boundary conditions are applied in the spanwise direction while no-slip conditions are imposed on the cylinder surface. Characteristic based conditions are used at the inflow and outflow as well as on the lateral surfaces. The freestream Mach number is set equal to 0.1 in order to make a sensible comparison with incompressible simulations in the literature. Preconditioning is used to deal with the low Mach number regime. The computational domain is discretized by an unstructured grid consisting of approximately $274K$ nodes. The simulation is performed using our hybrid RANS/VMS-LES model with a dynamical SGS closure. The treatment of the boundary layer has received much attention, in our case we capture it with the statistical model, RANS with $k-\varepsilon$ closure model and wall treatment with wall law as explained in Section 3.4.2. A schematic illustration of the boundary layer with wall law treatment is done in Figure 3.9, where:

- $\delta_{matching}$ is the thickness of the wall law zone, and we shall denote y^+_{match} the normalized distance y to the wall: $y^+_{match} = \frac{\delta_{match U_f}}{\nu}$
- y^+_{mesh} is the normalized size of the first layer of the mesh : $y^+_{mesh} = \frac{\Delta x \ U_f}{\nu}$,

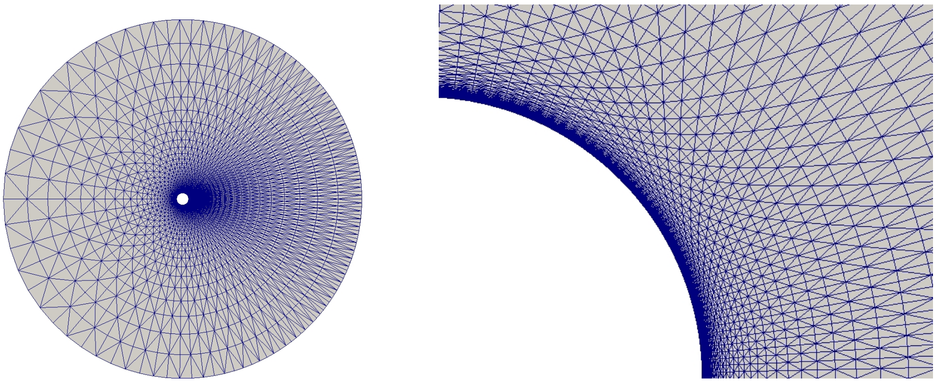


Figure 3.8: Snapshot of the surface mesh with $274K$ cells, $y+_{match} = 100$ and $y+_{mesh} = 10$ (image on the left); zoom on the near-wall mesh (image on the right).

where the value of the friction velocity U_f is a priori chosen within the empirical law: $U_f/U_\infty = 1/20$ (where U_∞ is the upstream velocity for dimensionless the friction velocity).

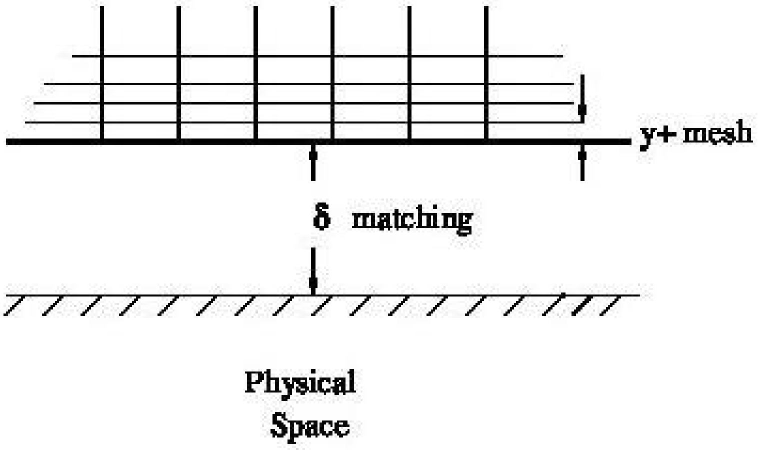


Figure 3.9: Sketch of the boundary treatment with wall law

As regards the blending function, the *protection zone* is used, with the blending function defined as in Relation (3.57) to avoid a mismatch near the wall. Figure 3.10 illustrates the blending function behavior for this benchmark at several Reynolds numbers, from $1.5M$ to $3M$. We observe with red color the RANS flow and in blue the LES flow part. The near wall region is indeed treated with RANS, and this layer diminishes with Reynolds number increasing. We observe also additional fluctuations at higher Reynolds number, both in the near wall and wake regions. As expected, the detachment angle decreases with Reynolds number and this is well captured by the blending function too.

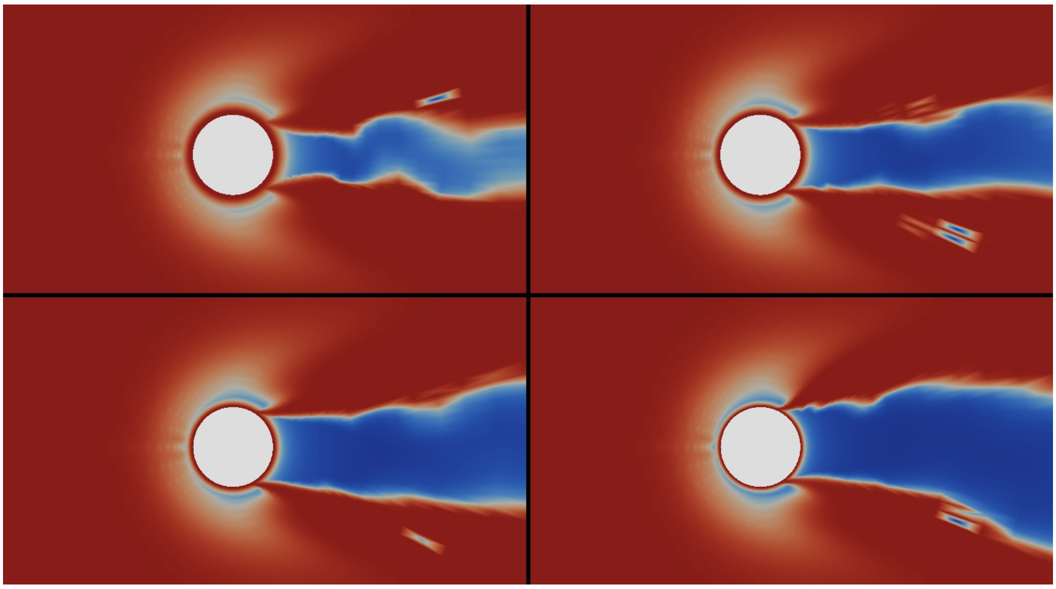


Figure 3.10: Hybrid computation: Visualisation of the instantaneous blending function (red part is RANS, blue one is VMS-LES) for Reynolds numbers: $1M$ (top left), $1.5M$ (top right), $2M$ (bottom left) and $3M$ (bottom right).

Computations are performed with our parallel MPI in-house solver AIRONUM written in Fortran 95 using 64 processors in this particular case. Several issues are adressed for this benchmark.

3.7.1 Maximum mixing length in the dynamical model

Firstly, regarding our dynamic SGS Smagorinsky model. In Figure 3.11 the effect of $(C_s\Delta_{max})^2$ parameter is analysed. We recall that C_s is the Smagorinsky constant and Δ is the size of the filter. The quantity $(C_s\Delta_{max})^2$ is interpreted as the maximum "mixing length". More precisely, we analyse the sensitivity of the computation to this filter component, usually in the litterature users allow for hundred times greater values than for the non-dynamic model. In our case according to our simulation shown in Figure 3.11 when this parameter is twice compared to the size of the non-dynamic value, *i.e.*

$$C_s\Delta_{max} = 2 * C_s\Delta$$

then it tends to be too dissipative. In contrast, when we set it as the size of the non-dynamic:

$$C_s\Delta_{max} = C_s\Delta$$

we can say that it is a very reasonable compromise. Therefore, we choose the latter value for our simulations.

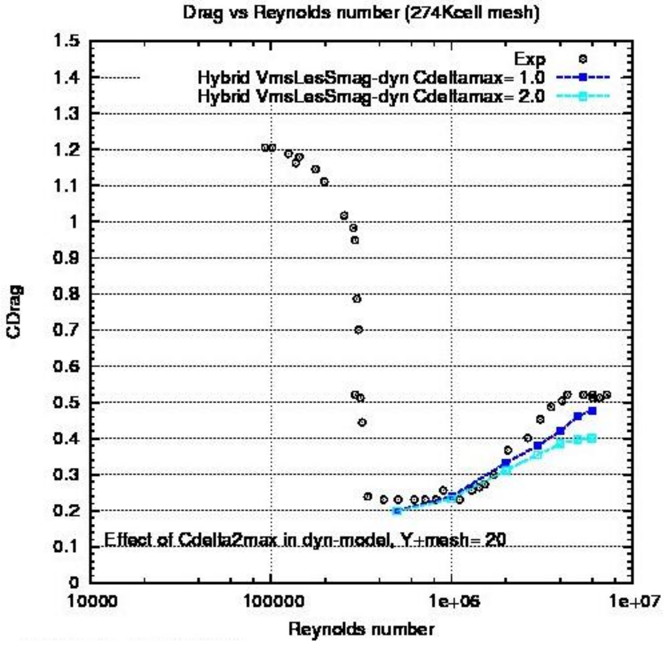


Figure 3.11: Effect of $C\Delta_{max}$ on the drag of our dynamic hybrid RANS/VMS-LES model for Reynolds numbers ranging from $500K$ to $8M$, comparison with experiments [Schewe 1983]

3.7.2 Analysis of the results

Drag and lift oscillations are captured for several Reynolds numbers: $1M$ (Figure 3.12), $1.5M$ (Figure 3.13) for our dynamical hybrid model. We can observe in these two figures that the mean drag, and lift oscillations, are growing with Reynolds number, which is the behaviour we expect for supercritical flows following the drag to Reynolds curve [2].

Since the grid we are using is quite coarse, of about $274K$ cells (see snapshot in Figure 3.8) it is reasonable to compare the hybrid computation with the statistical RANS one. In Figures 3.14 and 3.15 we perform this comparison for Reynolds numbers $2M$ and respectively $3M$. We observe that at such high Reynolds numbers, on the contrary to what we are used, the RANS flow have a very irregular behaviour, and at Reynolds=$3M$, the mean values are slightly larger than those find with the hybrid model (see also Tables 3.3, 3.4, 3.5, 3.6).

In Figure 3.16 we plot the drag coefficient at several supercritical Reynolds numbers for RANS and hybrid computations (using the same mesh) and compared with the experimental data. The simulation shows that starting from $Re = 3M$ our hybrid model still follows the experiments, while the RANS simulation gives a drag

[2] The experimental data were taken from Figure 2, p. 289 of [Schewe 1983]. Some experimental data values are missing and reflects the difficulty of picking values from a figure.

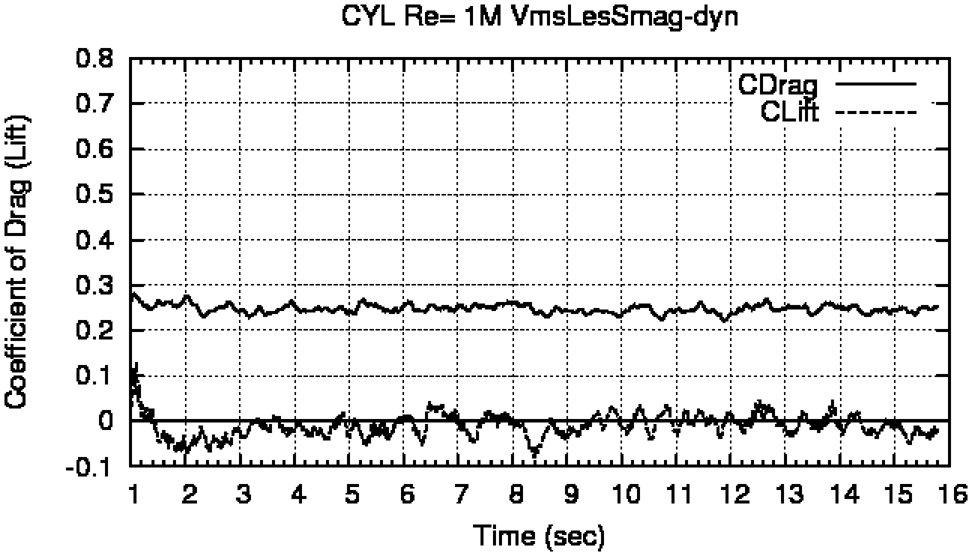


Figure 3.12: Flow over a circular cylinder at Reynolds number $1M$: Time-dependent forces. Upper curve: drag C_d, lower curve: lift C_l.

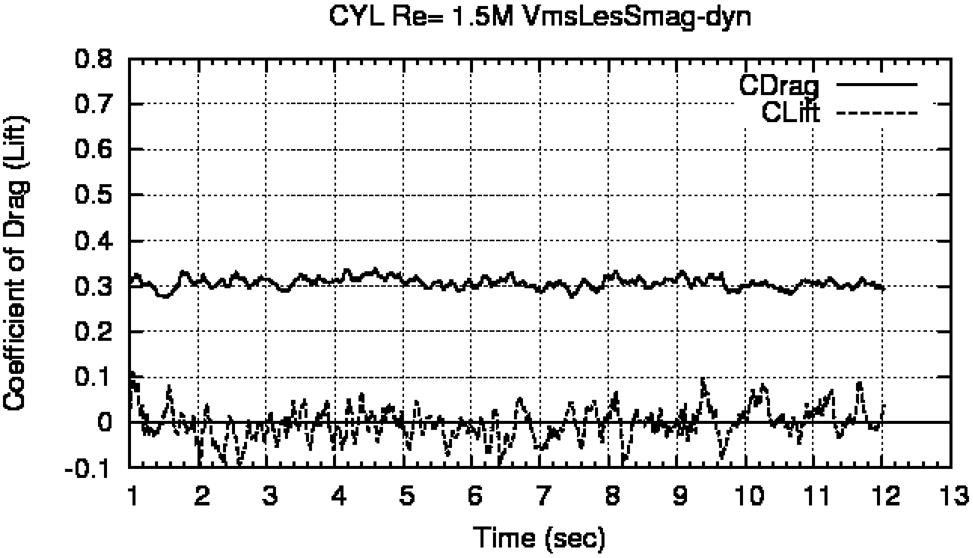


Figure 3.13: Flow over a circular cylinder at Reynolds number $1.5M$: Time-dependent forces. Upper curve: drag C_d, lower curve: lift C_l.

value above the experimental one.

Next, the instantaneous vorticity is plotted in Figure 3.17 for hybrid computation for several Reynolds number Rey: $1M$, $1.5M$, $2M$ and $3M$. Comparing these results we notice that the wake is narrower at smaller Rey and the boundary-layer separation slightly delayed than for $Rey = 3M$, resulting in a slightly smaller drag

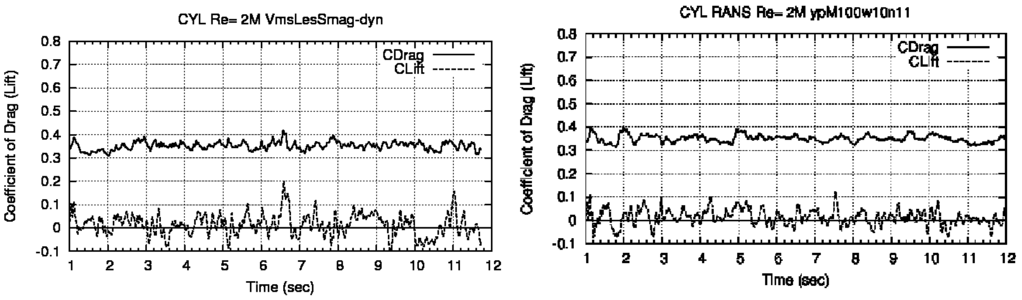


Figure 3.14: Flow over a circular cylinder at Reynolds number $2M$: Time-dependent forces. Upper curve: drag C_d, lower curve: lift C_l. Comparison between dynamic hybrid RANS/VMS-LES model (left image) with RANS (right image)

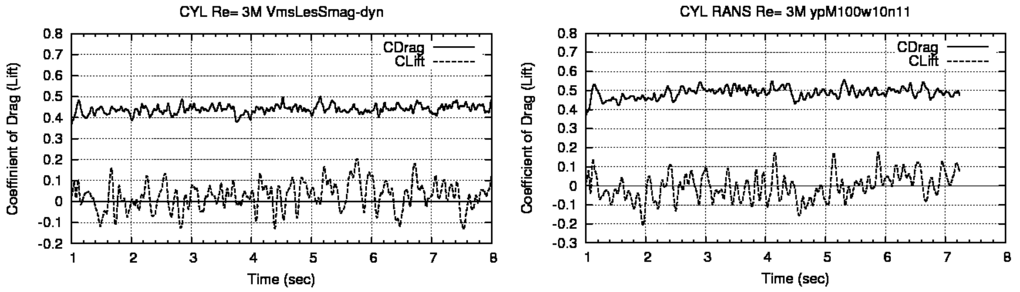


Figure 3.15: Flow over a circular cylinder at Reynolds number $3M$: Time-dependent forces. Upper curve: drag C_d, lower curve: lift C_l. Comparison between dynamic hybrid RANS/VMS-LES model (left image) with RANS (right image)

coefficient. Note that the thin boundary layer seen in this Figure is not representative, the true boundary layer with strong vorticity is extremely thin and not captured here.

Aknowledgement
CINES (Montpellier, France), IDRIS (Orsay, France) and CINECA (Bologna, Italy) are gratefully acknowledged for having provided the computational resources.

3.7.3 Conclusion

Comparing with other hybrid turbulence methods in the literature the main orginality of our previously introduced hybrid approach comes from it's capability to integrate the VMS-LES formulation of the LES component. Our turbulence model presented here adjusts the SGS viscosity not only in spatial direction thanks to the dynamical model, but also according to fluctuation scales thanks to the VMS formulation. The simulations performed with our hybrid model shows a rather good agreement with experimental data for coarse meshes. For higher Reynolds numbers, the advantage of hybrid modelling with respect to RANS is observed. In contrast to LES calculation existing in the litterature, the Reynolds influence on drag is rather

Simulations	$\overline{C_d}$	C_l'	Cp_{base}	θ
Hybrid RANS/VMS-LES dynamic	**0.24**	**0.020**	**0.30**	117
RANS	**0.25**	**0.026**	**0.34**	**119**
LES of [Wang 2001]	**0.31**		**0.32**	
Experiments				
[Shih 1993]	**0.20**		**0.33**	
[Zdravkovich 1997]	.17±.40			
[Schewe 1983]	**0.22**			

Table 3.3: Bulk flow parameters prediction for simulations at $Rey = 1M$: $\bar{C}_d$ is the mean drag, C_l' is the root mean square (r.m.s) of the lift coefficient, θ is the separation angle. Comparison with experimental data of [Shih 1993, Zdravkovich 1997].

Simulations	$\overline{C_d}$	C_l'	Cp_{back}	θ
Hybrid RANS/VMS-LES dynamic	**0.30**	**0.034**	**0.36**	116
RANS	**0.31**	**0.044**	**0.38**	**124**
Experiments				
[Schewe 1983]	**0.22**			

Table 3.4: Bulk flow parameters prediction for simulations at $Rey = 1.5M$: $\bar{C}_d$ is the mean drag, C_l' is the root mean square (r.m.s) of the lift coefficient, θ is the separation angle.

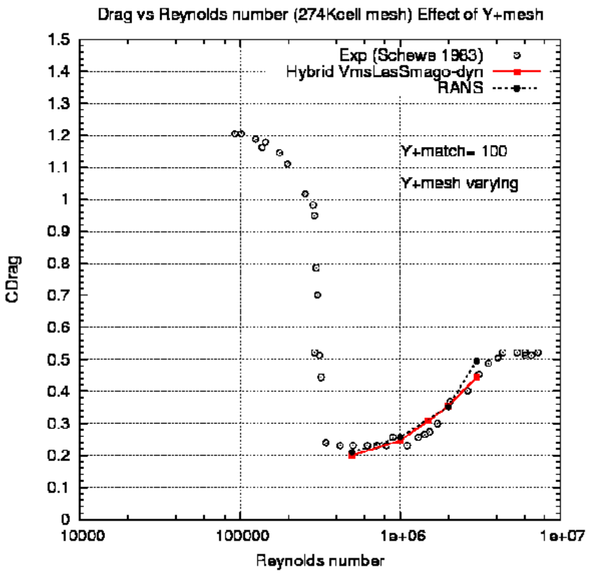


Figure 3.16: Drag vs. Reynolds curve: Compairing RANS and Hybrid computations with Experimental data of [Schewe 1983].

well captured. Further investigation on the sensitivity of the computation to y^+_{mesh}

Simulations	$\overline{C_d}$	C_l'	Cp_{back}	θ
Hybrid RANS/VMS-LES dynamic	**0.35**	**0.048**	**0.38**	**114**
RANS	**0.35**	**0.030**	**0.42**	**108**
LES [Wang 2001]	**0.32**	—-	—-	—-

Table 3.5: Bulk flow parameters prediction for simulations at $Rey = 2M$: $\bar{C_d}$ is the mean drag, $C_l^{'}$ is the root mean square (r.m.s) of the lift coefficient, *theta* is the separation angle.

Simulations	$\overline{C_d}$	C_l'	Cp_{base}	θ
Hybrid RANS/VMS-LES dynamic	**0.44**	**0.063**	**0.47**	**107**
RANS	**0.49**	**0.067**	**0.55**	**109**
DES with rotation/curvature terme [Travin 1999]	**0.51**	**0.10**	**0.64**	**106**
DES without rotation/curvature terme [Travin 1999]	**0.41**	**0.06**	**0.53**	**111**
Experiments [van Nunen 1974]	—	—-	——	**120**

Table 3.6: Bulk flow parameters prediction for simulations at $Rey = 3M$: $\bar{C_d}$ is the mean drag, $C_l^{'}$ is the root mean square (r.m.s) of the lift coefficient. comparison with experimental data of [van Nunen 1974].

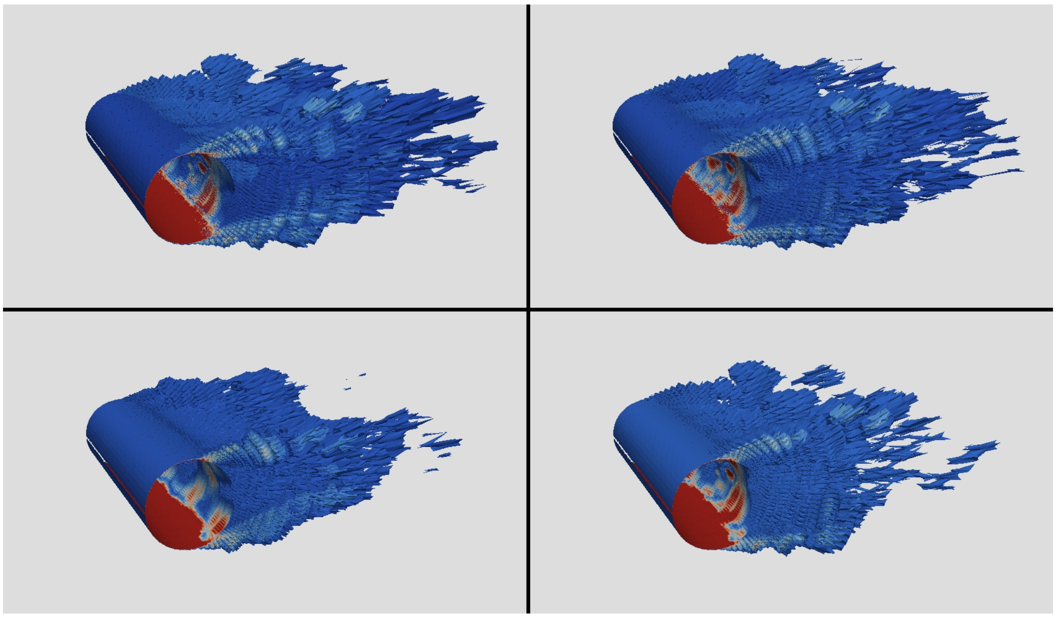


Figure 3.17: Hybrid computation: Visualisation of the instantaneous vorticity magnitude for Reynolds numbers $1M$ (top left), $1.5M$ (top right), $2M$ (bottom left) and $3M$ (bottom right).

and an improved theoretical law defining a relation between the size of the matching zone and the turbulence Reynolds number for supercritical flows, can be considered in the future.

Chapter 4

Continuous and discrete adjoint systems

Contents

4.1 Adjoint Methods - State of the Art

The use of adjoint states goes back to the introduction (probably by Lagrange) of Lagrange multipliers for Lagrangian fonctions defining dynamical systems. In his control theory in early 1962, Pontryagin [L.S. Pontryagin 1962] use adjoints for his Hamiltonian terms of duality. Next, in 1968 J-L. Lions [Lions 1968] built the Optimal Control theory for systems governed by partial differential equations with an extensive use of the adjoint notion. Ever since, the use of adjoint states has been spread to several disciplines who use the search for an optimum (minimum) under an equation constrained:

- Control
- Optimum Design
- Identification and Filtering
- Data assimilation in meteorology.

Design optimisation. Finding the best shape of an airfoil and of an aircraft is a task not well solved by intuition and, as soon as the other mathematical techniques (CFD, ...) have permit it, it was interesting to try to apply the Control Theory to the optimisation of the aerodynamical shape. The first use of adjoints based on output functionals for design is due to Pironneau [Pironneau 1974]. The equation of the state used by Pironneau and co-workers was the Potential equation for compressible flows. Then Jameson [Jameson 1987] was also one of the pioneers on the use of adjoint equations to develop what he calls optimal control methods for designing shapes, this time with the Euler equation as state system. These methods refers to the fact that one tries to find the geometry that minimises some objective functionals subject to a set of constraints. In other words, the heart of the algorithm consists in an optimisation procedure which uses an adjoint method to compute the linear sensitivity of the objective function with respect to a number of design variables.

The usual approach for the optimal design adjoint problem starts with the assumption that W_h is the discrete flow solution that satisfies the discrete system of equations:

$$\Psi_h(\gamma, W_h(\gamma)) = 0,$$

and γ is a single design variable.
Differentiating with respect to γ gives:

$$\frac{\partial \Psi_h}{\partial W_h}\frac{\partial W_h}{\partial \gamma} + \frac{\partial \Psi_h}{\partial \gamma} = 0$$

which determines the changes in W_h due to a change in γ. Given now an objective functional, $J_h(\gamma, W_h(\gamma))$, which could be for example, a discrete approximation of lift or drag of an airfoil, the change of this functional subject to changes in the design variable γ is given by:

$$\begin{aligned}\frac{dJ_h}{d\gamma} &= \frac{\partial J_h}{\partial W_h}\frac{dW_h}{d\gamma} + \frac{\partial J_h}{\partial \gamma}\\ &= -\frac{\partial J_h}{\partial W_h}\left(\frac{\partial \Psi_h}{\partial W_h}\right)^{-1}\frac{\partial \Psi_h}{\partial \gamma} + \frac{\partial J_h}{\partial \gamma}\\ &= W_h^{*,T}\frac{\partial \Psi_h}{\partial \gamma} + \frac{\partial J_h}{\partial \gamma},\end{aligned}$$

where W_h^* is the solution of the adjoint equation:

$$\left(\frac{\partial \Psi_h}{\partial W_h}\right)^T W_h^* + \left(\frac{\partial J_h}{\partial W_h}\right)^T = 0. \tag{4.1}$$

The key point in using adjoint equation in this case is that when one is interested in more that one design variable, the derivative of the functional with respect to these design variables is expressed in a similar manner, using the same adjoint flow solution. The only additional computation for each additional design variable is the evaluation of $\frac{\partial \Psi_h}{\partial \gamma}$ and its vector dot product with W_h^*. Each of these two steps involves minimal computational efforts, and so we can say that the overall computational cost is almost independent of the number of design variables.

Of course we cannot talk about sensitivity evaluation in numerical software without mentioning the concept of Automatic Differentiation (or **AD**).

Automatic Differentiation. Automatic Differentiation is a technique that starts from a code, for example a code computing $\Psi_h(W_h)$ and automatically generates codes to compute:

$$\frac{\partial \Psi_h}{\partial W_h} U$$

also called *forward mode*, or it can compute:

$$\left(\frac{\partial \Psi_h}{\partial W_h}\right)^T V$$

in *reverse mode*.

We can cite numerous authors e.g. ([P. Hovland 1998, N. Gauger 2007, Martinelli 2010a]) who used Automatic Differentiation to generate adjoint CFD codes. There exists several AD tools in the litterature, ADIFOR [C.H. Bischof 1998], TAMC [Giering 1998], TAPENADE [Hascoët 2005], ODYSSEE [Faure 1998]. For a short explanation on the global functionality of these tools, I refer here to TAPENADE, the AD software developed in the Tropics Team at INRIA. The latests version deal with Fortran95 and with ANSI C ([Hascoët 2005, Hascoët 2008]). Figure 4.1 shows the architecture of TAPENADE. It is implemented mostly in JAVA except for the separate front-ends which can be written in their own languages. Front- and back-ends communicate with the kernel via an intermediate abstract language ("IL") that makes the union of the constructs of individual imperative languages. Notice also the clear separation between the general-purpose program analysis and the differentiation engine itself.

Nevertheless, the use of adjoint systems for state-constrained optimisation is just one example of their utility. Among linearisation methods, adjoint methods have demonstrated their efficiency in numerical error analysis and correction. This probably dates back to the error analysis of Aubin-Nitsche for finite-element approximation of elliptic problems. For numerical applications, Becker and Rannacher [Becker 2001], and then Giles and co-workers [Giles 1997, Giles 1999, Pierce 2000, Giles 2002a, Pierce 2004] showed the efficiency of adjoint systems for a better prediction of an output functional associated with a given system of PDE (partial differential equations). The adjoint based error correction method distinguishes from other approaches who may have little relation to errors in integral outputs of primary concern to the engineer. As an example,

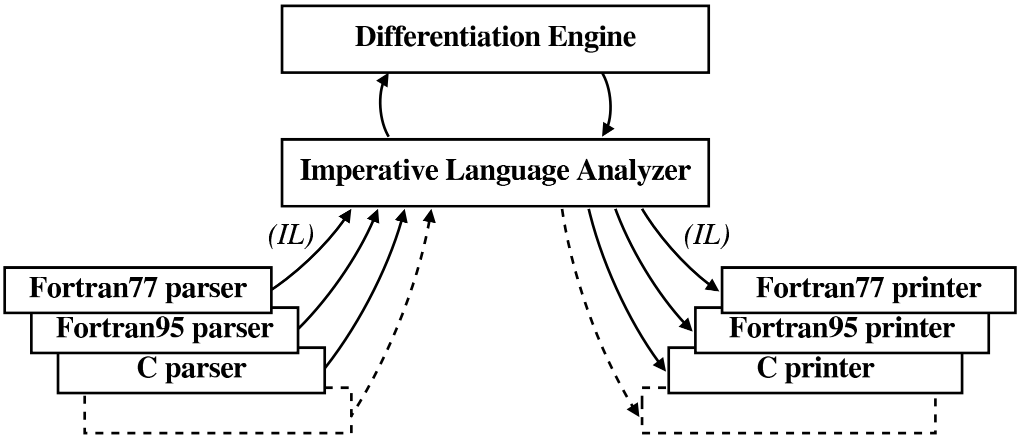


Figure 4.1: Overall Architecture of TAPENADE

consider the wake behind an aircraft plain. In order to adequately resolve the wake, a finer grid is required locally, but it is often the case that the computed wake located a chord or two downstream of the wing passes into a region in which the grid resolution is rather coarse. In this case, a grid adaptation based on an usual error estimation, for instance a truncation error, would probably induce further refinement in this region. However, the effort of refinement could be useless if errors in this region has a very small contribution to computed lift and drag. Refinement in region closer to the wing, probably near the leading and trailing edge would lead to much greater reduction in the lift and drag errors. As it will be shown in this thesis, adjoint based error estimators are a good indicator of local refinement when an output functional is being concerned.

4.2 Continuous Adjoint System

4.2.1 Continuous Adjoint for Euler system

We detail in this section the continuous adjoint system associated with unsteady inviscid Euler flows. The classical approach is based on the commun use of a Lagrangian from the optimisation theory. Another issue to express the continuous adjoint equation is by direct linearization of the state system, but we prefer the use of Lagrangian for a better understanding of the difficulties encountered in this developpement.

We recall the conservative expression of the compressible unsteady Euler equations:

$$\Psi(W) = W_t + \nabla.\mathcal{F}(W) = 0 \text{ on } \Omega$$

where W is the vector of conservative flow variables and $\mathcal{F}(W) =$

$(\mathcal{F}_1(W), \mathcal{F}_2(W), \mathcal{F}_3(W))$ is the usual Euler flux. We refer to Chapter 2 for a detailed description of Euler system of equations.
We impose a vanishing velocity in normal direction to the obstacle, that is: $\mathbf{u}\cdot\mathbf{n} = 0$ on Γ, where Γ is the solid boundary of the computational domain denoted here Ω and by $\mathbf{n}$ we understand the outward normal to Γ. We consider also a simplified boundary conditions on the far-field (Γ_∞): $W = W_\infty$.
For seek of simplicity we keep the same notation as in Chapter 5. We define thus a scalar functional: $j(W) = (g, W)$ which we observe on the solid boundary Γ.

The minimisation problem writes:

$$\min j(W) \text{ with } \Psi(W) = 0. \tag{4.2}$$

This optimisation problem can be transformed into an unconstraint minimisation problem with the introduction of a Lagrangian functional. Let us introduce the Lagrangian multipliers W^* defined on the computational domain Ω, $\bar{W}^*$ defined on part of the boundary Γ, and $\bar{\bar{W}}^*$ on the rest of the boundary. With this hypothesis we define the Lagrangian $\mathcal{L}$ as:

$$\begin{aligned}\mathcal{L}(W, W^*, \bar{W}^*, \bar{\bar{W}}^*) = &\int_0^T \int_\Gamma j(W)\mathrm{d}\Gamma\mathrm{d}t + \int_0^T \int_\Omega W^* \left(W_t + \nabla.\mathcal{F}(W)\right) \mathrm{d}\Omega\mathrm{d}t \\ &- \int_0^T \int_{\Gamma\backslash\Gamma_\infty} \bar{W}^*\mathbf{u}\cdot\mathbf{n}\mathrm{d}\Gamma\mathrm{d}t - \int_0^T \int_{\Gamma_\infty} \bar{\bar{W}}^*(W - W_\infty)\mathrm{d}\Gamma\mathrm{d}t\ .\end{aligned}$$

according to the Lagrange duality theory, the constrained minimum is obtained for a saddle point of $\mathcal{L}$, in our case when the following variation vanishes:

$$\begin{aligned}\delta_W\mathcal{L}(W, W^*, \bar{W}^*, \bar{\bar{W}}^*) = &\int_0^T \int_\Gamma \frac{\partial j}{\partial W}\delta W\mathrm{d}\Gamma\mathrm{d}t + \int_0^T \int_\Omega W^* \frac{\partial}{\partial W}\left(W_t + \nabla.\mathcal{F}(W)\right) \delta W\mathrm{d}\Omega\mathrm{d}t \\ &- \int_0^T \int_\Gamma \bar{W}^* \frac{\partial}{\partial W}\delta\mathbf{u}\cdot\mathbf{n}\ \mathrm{d}\Gamma\mathrm{d}t + \text{far-field contribution} \\ = &\int_0^T \int_\Omega \left(-W_t^* - \left(\frac{\partial\mathcal{F}}{\partial W}\right)^* \nabla W^*\right)\delta W\mathrm{d}\Omega\mathrm{d}t \\ &+ \int_0^T \int_\Gamma \left(\frac{\partial j}{\partial W} - \bar{W}^*\frac{\partial}{\partial W}\mathbf{u}\cdot\mathbf{n}\right)\delta W\mathrm{d}\Gamma\mathrm{d}t \\ &+ \int_0^T \int_{\Gamma\cup\Gamma_\infty} W^*\frac{\partial\mathcal{F}}{\partial W}\delta W\cdot\mathbf{n}\ \mathrm{d}\Gamma\mathrm{d}t + \text{far-field contribution.}\end{aligned}$$

Then the necessary conditions for minimum in W direction are obtained by requiring the integrands in the Lagrangian variation to vanish. This gives us the adjoint system and the corresponding adjoint boundary conditions:

$$-W_t^* - \left(\frac{\partial\mathcal{F}}{\partial W}\right)^* \nabla W^* \ = \ 0 \ \text{ on } \Omega \quad \textit{"Continuous Adjoint System"} \tag{4.3}$$

$$\frac{\partial j}{\partial W} + W^* \frac{\partial \mathcal{F}}{\partial W} \cdot \mathbf{n} - \bar{W}^* \frac{\partial}{\partial W} \left(\mathbf{u} \cdot \mathbf{n}\right) = 0 \text{ on } \Gamma \quad \textit{"Boundary Condition"} \tag{4.4}$$

From now, the adjoint far-field boundary conditions are not mentioned in our study, they are of less importance because no major difficulties is caused on the far-field. Regarding the adjoint system we observe a system of advection equations, where the temporal integration goes backwards, *i.e.,* in the opposite direction of usual time. Thus, when solving the unsteady adjoint system, one starts at the end of the flow run and moves back until reaching the start time. For unsteady problem this represents a major difficulty, even if several issues are proposed in the litterature to avoid storage and re-computation problems, this remains one source of the industrials skepticism regarding the use of adjoints in CFD commercial codes.

4.2.2 Continuous Adjoint Navier-Stokes system

We discuss in this section the continuous adjoint Navier-Stokes system for laminar viscous flows. We recall the compressible Navier-Stokes system in conservative formulation:

$$\Psi(W) = W_t + \nabla\ .\mathcal{F}(W) - \nabla\ .\mathcal{V}(W) \ = \ 0 \tag{4.5}$$

where the vector of conservative flow variables W, Euler flux $\mathcal{F}(W)$ and viscous flux $\mathcal{V}(W)$ are defined in Chapter 2. Equation (4.5) is supplemented with characteristic-type boundary condition on the far-field and non-slip adiabatic conditions on the solid wall Γ:

$$\mathbf{u} \ = \ 0 \ \text{ and } \ \frac{\partial T}{\partial \mathbf{n}} \ = \ 0. \tag{4.6}$$

The Lagrangian expression for Navier-Stokes system writes then:

$$\begin{aligned} \mathcal{L}(W, W^*, \bar{W}^*) = & \int_0^T \int_\Gamma j(W) \mathrm{d}\Gamma \mathrm{d}t + \int_0^T \int_\Omega W^* \left(W_t + \nabla.\mathcal{F}(W) - \nabla.\mathcal{V}(W)\right) \mathrm{d}\Omega \mathrm{d}t \\ & - \int_0^T \int_\Gamma \bar{W}^* \mathbf{u} \ \mathrm{d}\Gamma \mathrm{d}t - \int_0^T \int_\Gamma \bar{\bar{W}}^* \frac{\partial T}{\partial \mathbf{n}} \mathrm{d}\Gamma \mathrm{d}t + \mathrm{FFT}\ . \end{aligned}$$

By FFT we denoted the Far-Field Terms contribution, that are not discussed here.

Consequently, the variation of the Lagrangian writes:

$$\begin{aligned}
\delta_W \mathcal{L}(W, W^*, \bar{W}^*) = & \int_0^T \int_\Omega W^* \frac{\partial}{\partial W} \left(W_t + \nabla.\mathcal{F}(W) - \nabla.\mathcal{V}(W) \right) \delta W \mathrm{d}\Omega \mathrm{d}t \\
& + \int_0^T \int_\Gamma \frac{\partial j}{\partial W} \delta W \mathrm{d}\Gamma \mathrm{d}t - \int_0^T \int_\Gamma \bar{W}^* \frac{\partial \mathbf{u}}{\partial W} \mathrm{d}\Gamma \mathrm{d}t \\
& - \int_0^T \int_\Gamma \bar{\bar{W}}^* \frac{\partial}{\partial W} \left(\frac{\partial T}{\partial \mathbf{n}} \right) \mathrm{d}\Gamma \mathrm{d}t + \mathrm{FFT} \\
& = \int_0^T \int_\Omega \left(-W_t^* - \left(\frac{\partial \mathcal{F}}{\partial W} - \frac{\partial \mathcal{V}}{\partial W} \right)^* \nabla W^* \right) \delta W \mathrm{d}\Omega \mathrm{d}t \\
& + \int_0^T \int_{\Gamma \cup \Gamma_\infty} W^* \left(\frac{\partial \mathcal{F}}{\partial W} - \frac{\partial \mathcal{V}}{\partial W} \right) \delta W \cdot \mathbf{n} \; \mathrm{d}\Omega \mathrm{d}t \\
& + \int_0^T \int_\Gamma \left(\frac{\partial j}{\partial W} - \bar{W}^* \frac{\partial \mathbf{u}}{\partial W} - \bar{\bar{W}}^* \frac{\partial}{\partial W} \left(\frac{\partial T}{\partial \mathbf{n}} \right) \right) \delta W \mathrm{d}\Gamma \mathrm{d}t + \mathrm{FFT}.
\end{aligned}$$

Then the necessary conditions for minimum are obtained by requiring the integrand in the Lagrangian variation to vanish. This gives us the continuous adjoint Navier-Stokes system:

$$-W_t^* - \left(\frac{\partial \mathcal{F}}{\partial W} - \frac{\partial \mathcal{V}}{\partial W} \right)^* \nabla W^* \;=\; 0 \;\; \text{on } \Omega \qquad \textit{"Continuous Adjoint System"} \tag{4.7}$$

and the corresponding *Adjoint Boundary Conditions*:

$$\frac{\partial j}{\partial W} + W^* \left(\frac{\partial \mathcal{F}}{\partial W} - \frac{\partial \mathcal{V}}{\partial W} \right) \cdot \mathbf{n} - \bar{W}^* \frac{\partial \mathbf{u}}{\partial W} - \bar{\bar{W}}^* \left(\frac{\partial}{\partial W} \left(\frac{\partial T}{\partial \mathbf{n}} \right) \right) = 0 \text{ on } \Gamma. \tag{4.8}$$

4.3 Discrete Adjoint System

We present in this section the discrete approach for deriving the adjoint equation, for both inviscid and viscous flows. This derivation remains as for continuous adjoint, i.e. based on a Lagrangian formulation. By discrete approach we understand that state system is first discretised in space and time and then integrated in the discrete Lagrangian formulation for transposition. We denote h the space-discretisation and n the temporal one. We restrict to the case of a first order Runge-Kutta scheme for sake of simplicity, but higher order scheme can be used, either not practical for the discret formulation. The developpement for implicit time advancing scheme is similar, although, in this later case, an extra matrix-vector product appears in the right-hand side of the system.

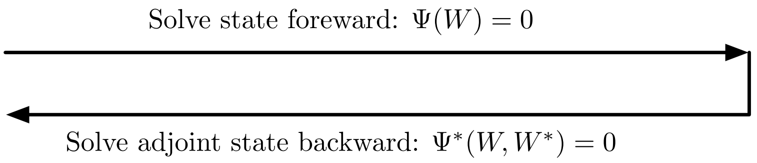

Solve state foreward: $\Psi(W) = 0$
Solve adjoint state backward: $\Psi^*(W, W^*) = 0$

Figure 4.2: Sketch of state-adjoint resolution.

4.3.1 Discrete Adjoint Euler System

Consider the following semi-discrete unsteady compressible Euler model (explicit RK1 time integration):

$$\Psi_h^n(W_h^n, W_h^{n-1}) = \frac{W_h^n - W_h^{n-1}}{\delta t^n} + \Phi_h^{\mathcal{F}}(W_h^{n-1}) = 0 \quad \text{for } n = 1, ..., N. \tag{4.9}$$

The time-dependent functional is discretized as follows:

$$j(W_h) = \sum_{n=1}^{N} \delta t^n j^{n-1}(W_h).$$

The problem of minimizing the error committed on the target functional $j(W_h) = (g, W_h)$, subject to the Euler system (8.14), can be transformed into an unconstrained problem for the following Lagrangian functional:

$$\mathcal{L} = \sum_{n=1}^{N} \delta t^n j^{n-1}(W_h) - \sum_{n=1}^{N} \delta t^n (W_h^{*,n})^T \Psi_h^n(W_h^n, W_h^{n-1})$$

where $W_h^{*,n}$ are the N vectors of the Lagrange multipliers (which are the time-dependent adjoint states). The conditions for an extremum becomes then:

$$\frac{\partial \mathcal{L}}{\partial W_h^{*,n}} = 0 \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial W_h^n} = 0, \quad \text{for } n = 1, ..., N.$$

The first condition is clearly verified from relation (8.14). Thus the Lagrangian multipliers $W_h^{*,n}$ must be chosen such that the second condition of extrema $\frac{\partial \mathcal{L}}{\partial W_h^n} = 0$ is verified. This gives the unsteady discrete adjoint system:

$$\left\{ \begin{array}{rcl} W_h^{*,N} & = & 0 \\ W_h^{*,n-1} & = & W_h^{*,n} + \delta t^n \dfrac{\partial j^{n-1}}{\partial W_h^{n-1}} - \delta t^n (W_h^{*,n})^T \dfrac{\partial \Phi_h^{\mathcal{F},n-1}}{\partial W_h^{n-1}} \end{array} \right. \tag{4.10}$$

As the adjoint system runs in reverse time, the first expression in the adjoint system (8.15) is referred to as adjoint "initialization".

Computing $W_h^{*,n-1}$ at time t^{n-1} requires the knowledge of state W_h^{n-1} and adjoint state $W_h^{*,n}$. Therefore, the knowledge of all states $\{W_h^{n-1}\}_{n=1,N}$ is needed to compute backward the adjoint state from time T to 0 which involves large memory

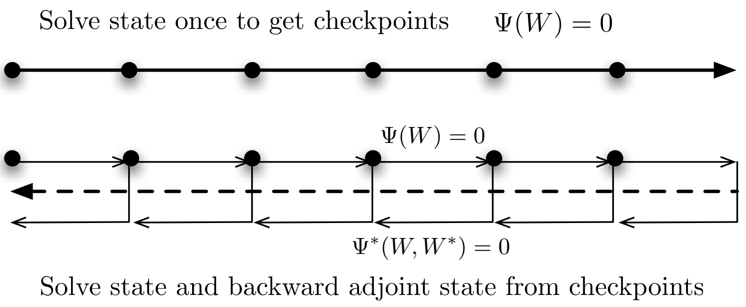

Solve state once to get checkpoints
$\Psi(W)=0$
$\Psi(W)=0$
$\Psi^*(W,W^*)=0$
Solve state and backward adjoint state from checkpoints

Figure 4.3: Checkpoint algorithm for discrete adjoint state computation

storage effort. This drawback can be reduced by out-of-core storage of checkpoints (as shown in the Picture 4.3), although it implies a recomputing effort of the state W. These methods are based on the idea of "divide and conquer". It consist in storing the state solution at some checkpoints and restarting the computation from this latest point. Several checkpointing methods exists in the litterature, for example in 1992 Griewank [Griewank 1992] proposed the scheme *Revolve*, capable to achieve optimal logarithmic behavior in terms of both computational time and memory requierement.

In this thesis the discrete adjoint formulation is prefered to the continuous one. One of the reasons is the consistency with the rest of our numerical algorithm.

4.3.2 Discrete Adjoint Navier-Stokes System

Similary to Euler flows, we present now the discret adjoint system associted to viscous laminar flows.

Let us consider the semi-discrete unsteady Navier-Stokes model. For Runge-Kutta first order scheme it writes:

$$\Psi_h^n(W_h^n, W_h^{n-1}) = \frac{W_h^n - W_h^{n-1}}{\delta t^n} + \Phi_h^{\mathcal{F}}(W_h^{n-1}) + \Phi_h^{\mathcal{V}}(W_h^{n-1}) = 0 \quad \text{for } n = 1, ..., N \qquad (4.11)$$

where by $\Phi_h^{\mathcal{F}}$ we denoted the flux contribution from convective (Euler) terms and respectively $\Phi_h^{\mathcal{V}}$ for viscous terms flux.

For seek of consistency we keep the same global notation for the functional:

$$j(W_h) = \sum_{n=1}^{N} \delta t^n j^{n-1}(W_h).$$

and the discrete Lagrangian:

$$\mathcal{L} = \sum_{n=1}^{N} \delta t^n j^{n-1}(W_h) - \sum_{n=1}^{N} \delta t^n (W_h^{*,n})^T \Psi_h^n(W_h^n, W_h^{n-1})$$

where $W_h^{*,n}$ are the N vectors of the Lagrange multipliers.

As previously explained, from the conditions of minimum we derive the discrete adjoint system, which for Navier-Stokes flow writes:

$$\left\{\begin{array}{rcl} W_h^{*,N} & = & 0 \\ W_h^{*,n-1} & = & W_h^{*,n} + \delta t^n \dfrac{\partial j^{n-1}}{\partial W_h^{n-1}} - \delta t^n (W_h^{*,n})^T \dfrac{\partial \Phi_h^{\mathcal{F},n-1}}{\partial W_h^{n-1}} - \delta t^n (W_h^{*,n})^T \dfrac{\partial \Phi_h^{\mathcal{V},n-1}}{\partial W_h^{n-1}} \end{array}\right.$$

Thus, for discrete adjoint Navier-Stokes system, an extra effort of computation is necessary, that is, the product between the implicit matrix associated to viscous flux with the adjoint vector: $(W_h^{*,n})^T \frac{\partial \Phi_h^{\mathcal{V},n-1}}{\partial W_h^{n-1}}$.
Regarding the algorithm of resolution, it is identical to the one explained in the previous section for Euler flows, based on checkpointing.

4.4 Example of discrete adjoint computation

In this section we present an example of adjoint computation for Euler flows. The test case represents a 2D blast into a city.

4.4.1 2D blast in a city: Adjoint Euler computation

The simulation of a blast in a 2D geometry representing a city is performed, see Figure 8.1. A blast-like initialization $W_{blast} = (10, 0, 0, 250)$ in ambient air $W_{air} = (1, 0, 0, 2.5)$ is considered in a small region of the computational domain.

We perform a forward/backward computation on a uniform mesh of 22 574 vertices and 44 415 triangles. Output functional of interest j is the quadratic deviation from ambient pressure on target surface S which is a part of the higher building roof (Figure 8.1):

$$j(W) = \int_0^T \int_S \frac{1}{2}(p(t) - p_{air})^2 \,\mathrm{d}S\,\mathrm{d}t.$$

Figure 8.2 plots the density isolines of the flow at different times showing several shock waves traveling throughout the computational domain. Figure 8.3 depicts the associated density adjoint state progressing backward in time. The same computational time is considered for both figures.

The simulation points out the ability of the adjoint to automatically provide the sensitivity of the flow field on the functional. Indeed, at early time of the simulation (top left picture), a lot of information is captured by the adjoint, *i.e.,* non-zero adjoint values. We notice that shock waves which will directly impact are clearly detected by the adjoint, but also shocks waves reflected by the left building which will be redirected towards surface S. At the middle of the simulation, the adjoint neglects waves that are traveling in the direction opposite to S and also waves that will not impact surface S before final time T since they won't have an influence on the cost functional. While getting closer to final time T (bottom right picture), the adjoint only focuses on the last waves that will impact surface S and ignores the rest of the flow.

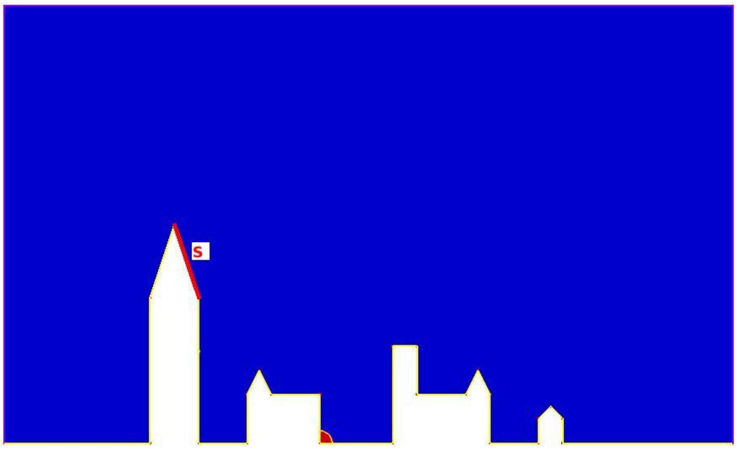


Figure 4.4: Initial blast solution and location of target surface S.

An example of adjoint state computation for Navier-Stokes flow is provided in Chapter 9.

4.5 Some remarks

From functional analysis theory, we could say that a continuous adjoint system can be easily derived for any functional output as far as the linearized system is well posed. These however doesn't mean that all functionals leads to properly defined adjoint boundary conditions. Several works in the litterature [Arian 1997, Castro 2007, A. Bueno-Orovio 2007, Anderson 1997] illustrate this problem and propose solutions usually by adding auxiliary boundary terms to Lagrangian definition. In [Castro 2007] it is concluded that for Euler flows only output functionals depending solely on the pressure can be considered and that for viscous flows, using the compressible Navier-Stokes equations, only functionals which involve the entire stress tensor are admissible. This area of functionals choice remains a deep challenge, but the problematic needs to be considered when working with continuous adjoint formulation. For discrete adjoint system this "admissibility" problem has not been reported.

In practice, starting from the well-posed adjoint boundary value problem, a numerical resolution method adapted to continuous adjoint equation need to be developed and integrated in the solver.

A central question remains open: **Which adjoint formulation should we use? Discrete or Continuous?**

The major drawbacks on using the continuous adjoint approach are:

- one must be sure that the derivative of the discrete functional can be properly defined since it is the gradient of the discretized continuous functional

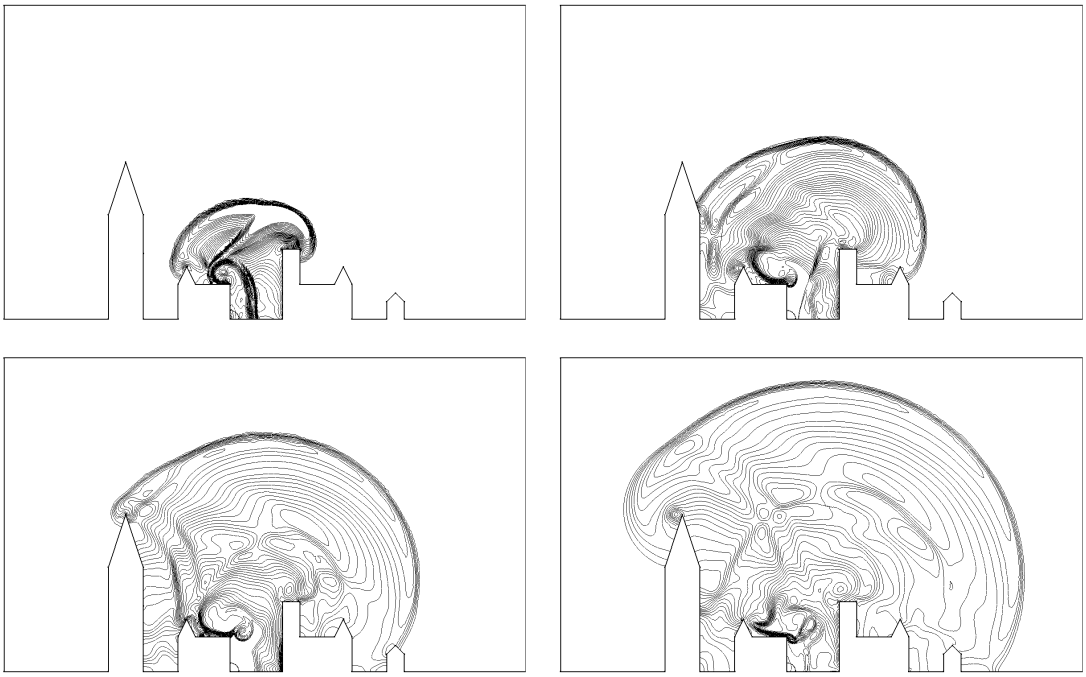


Figure 4.5: Density isolines evolving in time from left to right and top to bottom.

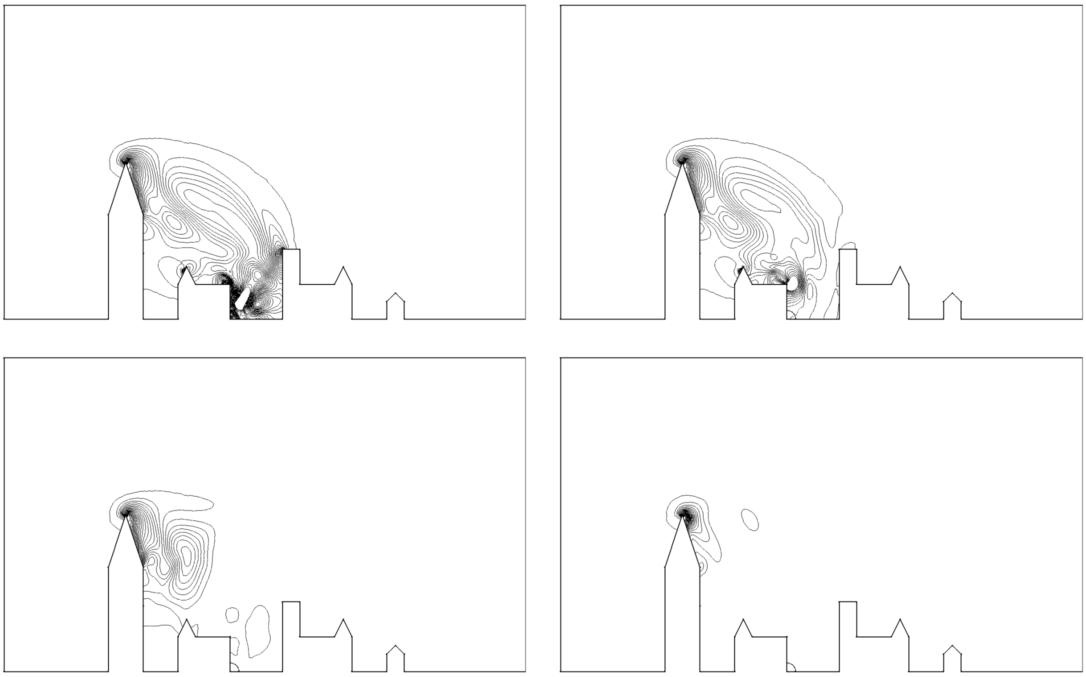


Figure 4.6: Density adjoint isolines evolving backward in time from left to right and bottom to top.

(generally if the mesh is sufficiently fine we can overcome this problem);

- the developpement of a numerical scheme adapted to adjoint equation can be fastidious; starting from the discretisation, choosing one that has the property of stability is sometimes a work a part, and same for higher order; a solution for example is to apply some Green formulas as explained in [Castro 2007];
- as explained earlier, we do not know yet how to generalize the adjoint boundary conditions due to some "inadmissible" functionals;
- another objection is that the continuous approach tends to neglect problems with some singularities (for example the ones issued from the geometry).

Regarding the discrete adjoint approach:

- it has the facility of understanding since working in $\mathbb{R}^n$, but the implementation of the exact gradient of a discrete functional is still difficult to achieve; in this context we are assisted by Automatic Differentiation tools;
- authors who adopted this approach noticed that the problems of linearization coming from discrete operators of differentiation are difficult to solve because they contain numerous non-linear terms issued from sophisticated schemes in finite volume method (approximate Riemann solvers, limiters).

A final answer to either use discrete or continuous formulation cannot be given. We suppose the choice will be governed by the context of its application. For our own purposes we adopted the discrete adjoint approach.

Chapter 5

Application to *a posteriori* error estimate

Contents

5.1 Introduction

This chapter presents a preliminary study performed for the last months of the European project NODESIM-CFD. NODESIM-CFD's subject is the management of uncertainties and errors in CFD simulations. Most part of the contribution of INRIA to NODESIM CFD concerns the management of random parameters with Taylor reduced-order models (see [Martinelli 2010b]). The second part of INRIA's contribution concerns a preliminary investigation in the management of adjoint-based numerical errors. The INRIA teams are expert in the construction and use of adjoint states. Adjoint states are rather complex to compute for steady CFD models. For unsteady CFD models, and in particular the 3D compressible models, the complexity is much higher. The main reason is that unsteady adjoint system solves backward in time and need to access to the solution of the forward in time state system, as it has been discussed in Chapter 4. Giles and Pierce [Pierce 2004] have proposed a method for estimating and correcting the error committed in computing a scalar functional defined from a state solution. They explain why a formulation based on adjoint is to be prefered to a formulation based on a linearised error state. This chapter presents this theory with some different notations and try to specify in which conditions the adjoint should be used and in which conditions it is better to prefer the linearised error state. The theory extends to unsteady model and we give a short numerical illustration.

In the first section of this chapter we present a short introduction of the main error estimators, then in Section 5.2 we focus on one of these estimators, that is *a posteriori* error analysis. Section 5.4 is dedicated to a particular component of the numerical error: numerical dissipation, and we conclude with a numerical example.

This small chapter on numerical error correction based on linearisation of the discrete state represents my contribution to NODESIM European Project and to my first paper in a journal [Martinelli 2010a].

5.2 Error estimates and correctors

Let us recall first how linearised -direct or adjoint- states can be useful for improving numerical accuracy issues.

The use of linearised systems in approximation of PDE allows deriving methods for managing approximation errors:

- by reducing the error via a corrector,
- by giving an error bound or estimate,
- by applying mesh adaptation controlled by an accurate error estimate.

The *numerical error* is the deviation between the solution $W = W(x, y, z)$ of mathematical model, for us a nonlinear PDE symbolized by:

$$\Psi(W) = 0 \ , \tag{5.1}$$

and the output data produced by the computations, i.e. the more or less perfect numerical solution that the computation will produce for solving the discrete system:

$$\mathbf{\Psi}_h(\mathbf{W}_h) = \mathbf{0} \ \in \ I\!R^N. \tag{5.2}$$

The discrete unknown $\mathbf{W}_h$ is the N-dimensional array of degrees of freedom:

$$\mathbf{W}_h \ \in \ I\!R^N \ , \ \mathbf{W}_h = [(\mathbf{W}_h)_i] \ .$$

Since W is a function and $\mathbf{W}_h$ an array, we cannot directly measure the deviation between them. The array $\mathbf{W}_h$ needs to be transformed into a function via an interpolation.

Let $V \ \in \ L^2(\Omega)$ be a space of rather smooth function (in practice, $V \ \subset \ \mathcal{C}^0(\bar{\Omega})$). Let R_h be a interpolation operator transforming an array of N degrees of freedom into a continuous function:

$$R_h : I\!R^N \ \rightarrow \ V \ \ \mathbf{v}_h \ \mapsto \ R_h\mathbf{v}_h. \tag{5.3}$$

Let us denote:

$$W_h(x,y,z) = (R_h \mathbf{W}_h)(x,y,z).$$

Similarly, we need an operator mapping continuous functions into arrays. Let T_h be this operator transforming a continuous function into an array of N degrees of freedom:

$$T_h : V \;\rightarrow\; I\!R^N \quad v \;\mapsto\; T_h v. \tag{5.4}$$

It can be useful, but not always necessary, to take T_h as the adjoint of R_h:

$$T_h = R_h^*. \tag{5.5}$$

The deviation between the PDE solution and the numerical one can be defined as $W - W_h$. In practice it consists mainly of *approximation errors*, of *algorithmic errors*, arising typically because iterative algorithms are not iterated infinitely, and of *round-off errors* due to the fact that the programm is run in floating point arithmetics. We discuss here mainly of approximation errors, although the other ones may be also addressed in part by the method studied here.

In order to evaluate the approximation error, two kinds of estimates can be applied:

A posteriori estimate:

$$\Psi(W) - \Psi(W_h) = -\Psi(W_h) \tag{5.6}$$

where $\Psi(W_h)$ is the continuous residual applied to discrete solution. Then:

$$W - W_h \approx -\big[\tfrac{\partial \Psi}{\partial W}\big]^{-1}\Psi(W_h). \tag{5.7}$$

A priori estimate:

$$\mathbf{\Psi}_h(T_h W) - \mathbf{\Psi}_h(\mathbf{W}_h) = -\mathbf{\Psi}_h(T_h W) \tag{5.8}$$

where $\Psi_h(T_h W)$ is the discrete residual applied to discretised continuous solution. Then:

$$T_h W - \mathbf{W}_h \approx -\big[\tfrac{\partial \mathbf{\Psi}_h}{\partial \mathbf{W}_h}\big]^{-1}\mathbf{\Psi}_h(T_h W). \tag{5.9}$$

In the *a posteriori* estimate (5.7), W_h can be either $W_h = R_h \mathbf{W}_h$, where $\mathbf{W}_h$ is solution of (5.2), or can be simply chosen arbitrarily. A particular case of the latter option is $R_h W_h'$, where W_h' is polluted by solution algorithm error, for example due to an insufficient iterative convergence. In contrast, in the *a priori* estimate (5.9), we assume that $\mathbf{W_h}$ is solution of (5.2).

We observe that these estimates involve unavailable continuous functions:
- In the *a posteriori* estimate (5.7), the solution of the continuous linearised system

is generally not available, it can be approximated thanks to the discrete Jacobian.
- For the *a priori* estimate, it is the T_hW term which we do not know. In Chapter 6 we propose an answer to this issue. It consists in replacing $\mathbf{\Psi}_h(T_hW)$ by an expression $T_h\Theta_h(W_h)$ depending only of W_h.

Let us concentrate on the *a posteriori* estimate. If T_hW_h is close to W in some norm, we expect that this implies that this norm is strong enough for ensuring that $\Psi(W_h)$ is also small enough. This assume a convergence of the residual $\Psi(W_h)$, i.e. something like a superconvergence W_h.

This kind of superconvergence has been obtained only for a few simplified contexts [Pierce 2000]. In the particular case of finite-difference method (FDM) on regular meshes, when a smooth interpolation (like a cubic spline) is applied, superconvergence is numerically evident. But a theory is missing for predicting it in a rigorous way.

Otherwise, superconvergence of derivatives (or of equation residual) can be obtained after some postprocessing. We refer for example to patch recovery methods as in the works of Zienkiewicz-Zhu [Zienkiewicz 1992a] and Krizek-Neittaanmaki [Krizek 1998].

Between (5.6), (5.8) and (5.7), (5.9) an approximation is made (with loss in accuracy) by a Taylor formula, using linearisation of the operators. This will allow us to identify a corrector which is (1) computable and (2) represents the dominant part of error. This can be done:

- either by linearisation of the PDE and writing a perturbed system,
- or by linearisation and transposition for writing an Adjoint system.

Comparaisons between both options can be performed in order to maximize the correctors efficiency.

5.3 *A posteriori* analysis issues

In this section we are particulary interested by an algorithm of linearisation for numerical error correction of a functional of observation following an *a posteriori* error analysis. We start from the theory proposed by Giles and co-workers [Giles 2002b, Pierce 2004, Giles 2002a].

Definition of correctors. We are thus interested on a better approximation for a scalar functional, in other words, starting from a supposed approximation to the true value, we "correct" this estimate by adding an extra term.
Let j be a smooth linear functional applying W into the scalar number:

$$j(W) = (g, W)_{L^2(\Omega)}$$

where g is a given $L^2(\Omega)$ function. The function $W \in L^2(\Omega)$ is assumed to be the solution of a nonlinear PDE describes by the continuous residual Ψ:

$$\Psi(W) = 0 \ . \tag{5.10}$$

The goal-oriented *a posteriori analysis* assumes that we have, instead of W, a function $\bar{W}$ and that we want to evaluate the deviation between $j(\bar{W})$ and $j(W)$. We have, applying some implicit function theorem:

$$j(W) - j(\bar{W}) = (g, W - \bar{W}) \approx -\left(\ g, (\frac{\partial \Psi}{\partial W})^{-1} \Psi(\bar{W}) \right) \ . \tag{5.11}$$

This allows to define:

$$\begin{aligned} \mathbf{g}_h &= T_h g \\ g_h &= R_h \mathbf{g}_h = R_h T_h g \end{aligned} \tag{5.12}$$

Let us introduce the continuous adjoint state W^*, solution of:

$$(\frac{\partial \Psi}{\partial W})^* W^* = g \ .$$

The discrete adjoint equation is then defined by:

$$\left[\tfrac{\partial \mathbf{\Psi}_h}{\partial \mathbf{W}_h}\right]^T \mathbf{W}^*{}_h = T_h g \ . \tag{5.13}$$

And we can then consider:

$$W_h^* = R_h \mathbf{W}^*{}_h \ .$$

A fundamental assumption of the present analysis is that this discrete adjoint is a good enough approximation of continuous adjoint W^* for allowing to replace W^* by W_h^* in the sequel.

Estimate 5.11 writes then:

$$j(W) - j(\bar{W}) = (W^*, \Psi(\bar{W})) \quad + \ R \ . \tag{5.14}$$

The notation R hold for a rest small enough to be neglected next. We shall say now that we can define a better approximation for the value of functional at W:

$$j(W) \ \approx \ j(\bar{W}) + (W^*, \Psi(\bar{W})) \ . \tag{5.15}$$

The right-hand side (RHS) of (5.15) is more accurate than $j(\bar{W})$ but its evaluation needs to know W^*, which is as difficult as knowing W. With the assumption that:

$$(W^* - W_h^*, \Psi(\bar{W})) \text{ is small enough} \ , \tag{5.16}$$

the corrected evaluation of j writes:

$$j(W) \ \approx \ j(\bar{W}) + (W_h^*, \Psi(\bar{W})) \ . \tag{5.17}$$

In practice, $\bar{W}$ is derived from solving an approximate system and for W_h^* we shall use also an approximate system.

We now want to correct or bound the error committed by using the above approximation for finding $\bar{W}$. This means that we put $\bar{W} = R_h\mathbf{W}_h$ and get the *adjoint corrector term*:

$$j(W) \ \approx \ (g, R_h\mathbf{W}_h) + (R_h\mathbf{W^*}_h, \Psi(R_h\mathbf{W}_h)\ )\ . \tag{5.18}$$

This can be also written as a *linearised direct corrector*:

$$j(W) \ \approx \ (g, R_h\mathbf{W}_h) - \langle T_h g, \left[\tfrac{\partial\mathbf{\Psi}_h}{\partial\mathbf{W}_h}\right]^{-1} R_h^*\Psi(R_h\mathbf{W}_h)\rangle\ . \tag{5.19}$$

Or equivalently, introducing the linearised quantity W_{lh}:

$$\begin{aligned} W_{lh} = R_h\mathbf{W}_h - T_h^*\left[\tfrac{\partial\mathbf{\Psi}_h}{\partial\mathbf{W}_h}\right]^{-1} R_h^*\Psi(R_h\mathbf{W}_h) \\ j(W) \ \approx \ (g, W_{lh})\ . \end{aligned} \tag{5.20}$$

Another way to correct directly the accuracy of $R_h\mathbf{W}_h$ is the Defect Correction method, which uses also the truncation error $\Psi(R_h\mathbf{W}_h)$. We first need to transfer this error into an array. Let us use T_h for this. The Defect Correction iteration consists in putting:

$$W_{dh} = R_h\left(\mathbf{W}_h - \left[\tfrac{\partial\mathbf{\Psi}_h}{\partial\mathbf{W}_h}\right]^{-1} T_h\Psi(R_h\mathbf{W}_h)\right)\ . \tag{5.21}$$

This gives the *defect corrector term*:

$$j(W) \ \approx \ (g, W_{dh}) = (g, R_h\mathbf{W}_h) - \langle R_h^* g, \left[\tfrac{\partial\mathbf{\Psi}_h}{\partial\mathbf{W}_h}\right]^{-1} T_h\Psi(R_h\mathbf{W}_h)\rangle\ . \tag{5.22}$$

This corrector is different from adjoint corrector (5.18-5.19). Giles and coworkers ([Pierce 2004]) observed that (5.22) can be a corrector less accurate that the adjoint one. However, our notations show that formulations (5.18-5.19) from one side and (5.22) from the other side are identical under the condition that

$$R_h^* = T_h\ . \tag{5.23}$$

This is the option that we recommand.
Remark 1: As observed by Giles *et al.* [Pierce 2004], both corrections, adjoint and defect, can be combined in order to get an even higher accuracy. For this we put:

$$j(W) \ \approx \ (g, R_h\mathbf{W}_{dh}) + (R_h\mathbf{W}_h^*, \Psi(R_h\mathbf{W}_{dh})\ )\ . \tag{5.24}$$

In case where Defect Correction is less accurate than (5.20), it may be interesting to replace $R_h\mathbf{W}_{dh}$ by W_{lh} in (5.24). □

Remark 2: Another way to exploit the Defect Correction principle is to evaluate the Defect not on the exact residual but on the residual of a higher order approximation:

assume that we can compute easily the residual of a fourth-order scheme for the vertex-centered node location:

$$\Psi_{4,h}(\bar{W}_h) = 0 \;\Rightarrow\; |W - \bar{W}_h| = O(h^4) \ , \tag{5.25}$$

then putting:

$$W_{d4h} = R_h \left(\mathbf{W}_h - \left[\tfrac{\partial \mathbf{\Psi}_h}{\partial \mathbf{W}_h}\right]^{-1} T_h \Psi_{4,h}(R_h \mathbf{W}_h) \right) \tag{5.26}$$

is another way to higher order accuracy. The central question is whether we can easily derive such a scheme and have a non cpu-costly evaluation of $\Psi_{4,h}(R_h\mathbf{W}_h)$. □

Remark 3
The computational efforts to compute the adjoint and the linearized corrector are very different. This is particularly true if we extend the above methods to the case of unsteady PDEs (addying time derivative inside the residual Ψ). Indeed, the linearised direct corrector is advanced simultaneously with the state. Conversly, the adjoint system has to be solved backward in time, while using the state solution at all time levels (see Chapter 8).

This remark leads to the following recommendations:

- use the direct linearised formulation in any case you only need a corrector for the field as well as a corrector for one or several output functionals.
- the adjoint formulation is compulsory when you wish to derive an goal-oriented optimal mesh.

The second recommendation is motivated by the fact that an optimal mesh will be derived from minimisation of the error term in which we need to put in evidence the dependance of error with respect to mesh. Since the adjoint is an approximation of a continuous function, it does not much depend of mesh. At the contrary, the continuous residual $T_h\Psi(W_h)$ or the truncation error $T_h\Theta_h(W_h)$ are proportional to a power of the mesh size. In Chapter 6 of this thesis, the truncation error is expressed in terms of adjoints, allowing the derivation (in Chapters 8 and 9) of an optimal mesh.□

Algorithms: In case where Relation (5.23) holds, we have two numerically equivalent algorithms to compute the (approximate) error-correction:

- *Adjoint method.* The algorithm writes as follows:
  - compute the N components of the vector $\mathbf{k}_h = R_h^* \Psi(W_h)$;

- compute the adjoint state $\mathbf{W}_h^* \in I\!R^N$ solution of the linear system

$$\left[\frac{\partial \mathbf{\Psi}_h}{\partial \mathbf{W}_h}\right]^T \mathbf{W}_h^* = \mathbf{g}_h \tag{5.27}$$

- compute the Euclidean dot-product $\langle \mathbf{W}_h^*, \mathbf{k}_h \rangle$.

- *Perturbative (adjoint-free) method:* this approach does not need the knowledge of the adjoint state, and this fact may be crucial for unsteady simulations. The algorithm writes as follow:

  - compute the N components of the vector $\mathbf{k}_h = R_h^* \Psi(W_h)$;
  - compute the vector $\mathbf{r}_h \in I\!R^N$ solution of the linear system

$$\left[\frac{\partial \mathbf{\Psi}_h}{\partial \mathbf{W}_h}\right] \mathbf{r}_h = \mathbf{k}_h \tag{5.28}$$

  - compute the Euclidean dot-product $\langle \mathbf{g}_h, \mathbf{r}_h \rangle$.

In the next section, we focus on a particular component of the error: *numerical dissipation*. This subject has been the object of numerous studies. Indeed, numerical dissipation is needed for stabilizing the approximation or to get rid of bad computed scales, like small eddies in large eddies simulations of turbulent flows. In the sequel we analyse a dissipation error correction model based on a linearisation of the discrete associated system and which re-uses some of its ingredients in such a way that the computational overcosts remains small. Two paradigms to specify the right-hand side of the linearized system can be considered. From one hand, the *a posteriori* estimator and, from the other hand, some particular terms whose target represents a particular part of the error (like the dissipation error).

5.4 Dissipation error correction

A posteriori error estimation methods have seen a major use on steady problems in conjunction with adjoint states usage allowing to focus on the accuracy of a given output functional. In this section we consider the use of linearized systems for the correction of the discrete flow itself. We recall first the results for a steady problem studied in [Martinelli 2010a]. Then we discuss the extension to unsteady systems. As long as adjoint state is not used, the computation of the flow and of the corrected one are performed in a quite similar manner. Two solutions of different accuracy or different models are advanced in time for each time step, the second one uses informations from the first one. This familly of methods has a large area of applications, from multiple discretisation formulations to *Non-Linear Disturbance Equations* for the coupling between turbulence models or aeroacoustic problems.

5.4.1 Corrected dissipation

We search for a solution $W = W(x, y, z, t)$ in a Hilbert space V of a PDE, thanks to a vector solution $(\mathbf{W}_h)_i \in I\!R^N$ of a discrete system:

$$\Psi(W) = 0 \quad ; \quad \mathbf{\Psi}_h(\mathbf{W}_h) = \mathbf{0} \in I\!R^N . \tag{5.29}$$

We then need the operators introduced in the previous section:

$$R_h : I\!R^N \rightarrow V \quad \mathbf{v}_h \mapsto R_h\mathbf{v}_h \;\; ; \;\; R_h^* : V \rightarrow I\!R^N . \tag{5.30}$$

for an accurate estimation of the error.

Let us recall the *a posteriori* estimate (5.7) defined in Section 5.2 :

$$\begin{aligned} W - W_h &\approx -\big[\tfrac{\partial\Psi}{\partial W}\big]^{-1}\Psi(W_h) \\ &\approx -R_h\big[\tfrac{\partial\mathbf{\Psi_h}}{\partial W_h}\big]^{-1}R_h^*\Psi(W_h). \end{aligned} \tag{5.31}$$

We refer to Section 5.2 of this chapter for details on this estimate.

Since we focus on correcting the numerical dissipation level, our proposed estimate writes then:

$$W_{h,corr} - W_h \approx -\big[\tfrac{\partial\Psi}{\partial W}\big]^{-1}(E_0(W_h) - E_1(W_h)) \tag{5.32}$$

where $E_1(W_h)$ is a dissipative and stable version of the advection operator and $E_0(W_h)$ is a non-dissipative but unstable operator.

5.4.2 A first example: Steady 2D Euler flow [Martinelli 2010a]

To illustrate the previous discussion, we give a first numerical example of corrector evaluation built on a finite-element approximation.

We can write steady Euler equations under the form:

$$\begin{gathered} W \in V = (H^1(Q) \cap \mathcal{C}^0)^5, \quad \forall\phi \in V, \\ \int_\Omega \mathcal{F}(W)\nabla\phi \, d\Omega - \int_{\partial\Omega} \phi\hat{\mathcal{F}}(W).n \, d\Gamma \;=\; 0\,. \end{gathered} \tag{5.33}$$

where $\hat{\mathcal{F}}(W)$ accounts for the different boundary conditions.
For the sake of simplicity we assume that the unknown W is H^1 and continuous in order to apply the usual interpolation operator.

Let us introduce a discretization of the previous EDP. Let τ_h a tetrahedrization of Ω with N vertices. It will rely on a discrete space of functions:

$$V_h = \{\phi_h \in H^1(\Omega)^5, \; \forall T \in \tau_h, \phi_h|T \in \mathcal{P}^1\}$$

the canonical basis of which is denoted:

$$V_h = span[N_i],\ N_i(x_j) = \delta_{ij} \forall i, j, \text{vertices of } \tau_h,$$

and on the interpolation operator:

$$\Pi_h : \mathcal{V} \to V_h,\ \Pi_h\phi(x_i) = \phi(x_i), \forall i, \text{vertex of } \tau_h.$$

Comparing with the previous abstract theory, we get:

$$\begin{array}{lll} R_h : & I\!R^{5N} \to V_h, & \mathbf{f}_h \mapsto R_h\mathbf{f}_h = \Sigma_i[\mathbf{f}_h]_i N_i\ , \\ T_h : & \mathcal{V} \to I\!R^{5N}, & \phi \mapsto T_h\phi = [\phi(x_i)]\ . \end{array}$$

The discretization is set into the discrete space, but also it differs from the continuous statement in two features, a discrete flux $\mathcal{F}_h$ instead of $\mathcal{F}$:

$$\mathcal{F}_h : \mathcal{V} \to V'$$

and an extra term of artificial diffusion D_h:

$$W_h \in V_h,\ \ \forall\phi_h \in V_h,\ (\Psi_h(W_h), \phi_h)_{V'\times V} = 0,$$

with

$$(\Psi_h(W_h), \phi_h)_{V'\times V} = \int_\Omega \phi_h \nabla.\mathcal{F}_h(W_h)\ \mathrm{d}\Omega - \int_\Gamma \phi_h \bar{\mathcal{F}}_h(W_h).n\ \mathrm{d}\Gamma + \int_\Omega \phi_h\, D_h(W_h)\mathrm{d}\Omega\,.$$

The discrete fluxes are chosen as follows:

$$\begin{aligned} \mathcal{F}_h(W) = \mathcal{F}_h(\Pi_h W) = \Pi_h\mathcal{F}(\Pi_h W)\,. \\ \bar{\mathcal{F}}_h(W) = \bar{\mathcal{F}}_h(\Pi_h W) = \Pi_h\bar{\mathcal{F}}(\Pi_h W). \end{aligned} \tag{5.35}$$

After some calculations and simplifications, the main error term appears as follows:

$$\begin{aligned} (\frac{\partial\Psi_h}{\partial W_h}\delta W_h, \phi_h) = & -\int_\Omega \nabla\phi_h(\mathcal{F}(W_h) - \Pi_h\mathcal{F}(W_h))\mathrm{d}\Omega \\ & + \int_\Gamma \phi_h(\hat{\mathcal{F}}^{out}(W_h) - \Pi_h\hat{\mathcal{F}}^{out}(W_h)).n\,\mathrm{d}\Gamma + \int_\Omega \phi_h\, D_h(W_h)\mathrm{d}\Omega \end{aligned} \tag{5.36}$$

with $\bar{\mathcal{F}}(W_h).n = \mathcal{F}(W_h).n - \hat{\mathcal{F}}(W_h).n$. A Gauss quadrature is applied for the evaluation of the right hand side. We have applied this to a steady subsonic flow and give some preliminary results (computation done by Martinelli cf. our common paper [Martinelli 2010a]). Figure 2 compares the entropy generation in the flow computed directly and the same flow corrected by formula (5.36). Entropy level is one order of magnitude smaller.

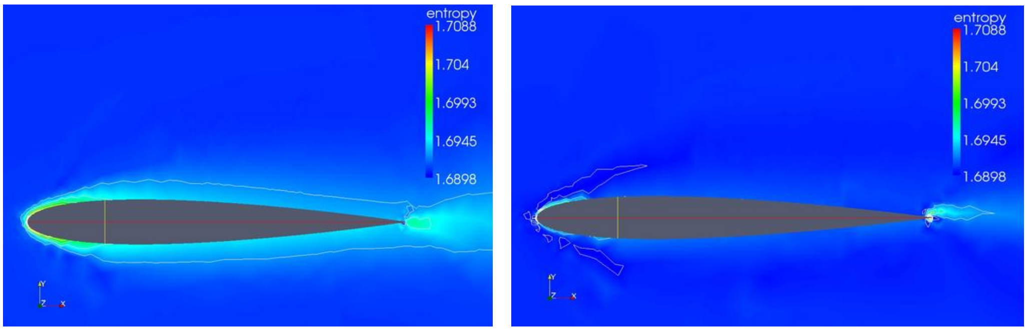


Figure 5.1: Entropy spurious generation for a direct computation of a steady flow (left image) and for a corrected one (right image)

5.4.3 Application to unsteady 3D Euler flows

Consider now the unsteady Euler equations:

$$\Psi(W) = W_t + \nabla \cdot \mathcal{F}(W) = 0 \quad \text{on} \quad \Omega.$$

Suppose the solution of this system regular enough, we have:

$$W \in \mathcal{V} = (H^1(Q) \cap \mathcal{C}^0)^5 \quad \text{and} \quad \forall\, \varphi \in \mathcal{V},\ (\Phi(W), \varphi)_{V' \times V} = 0.$$

We search for a discret solution:

$$\mathbf{W}_h,\ \text{such that}\ (\Psi_h(\mathbf{W}_h), R_h^* \phi_h) = 0, \forall \phi_h.$$

Our corrected solution needs to verify the *corrected system* writing:

$$\bar{\mathbf{W}}_h,\ \text{such that.}\ (\Psi_h(\bar{\mathbf{W}}_h), R_h^* \phi_h) = (\Psi(R_h \mathbf{W}_h), \phi_h), \forall \phi_h.$$

where by $\bar{\mathbf{W}}_h$ we denoted the corrected solution. The novelty is that we avoid the effort for linearisation, and maybe a part of the error coming from the linearisation.

Preliminary test for flow around a circular cylinder . The numerical dissipation error correction is applied for the flow around a circular cylinder at Reynolds number 20000. In order to install a solution on the considered three dimensional computational domain we use LES modelisation (see Chapter 3). The mesh is relatively coarse of about 43000 vertices.
After a purely LES computation, we restart the calculation with a new version of the programme which computes at each time step the flow and the one including the above correction. We compare the solution with the one obtained using the corrector. In Figure 5.2 is illustrated the RHS of the corrected system, which in this case is: $(E_0(W_h) - E_1(W_h))$. This term localizes numerical dissipation errors in this flow. We observe that error levels in the wake are higher. These errors have actually less impact on the forces that the liquid applies on the cylinder, but

they could impact the prediction of the low-frequency noise produced by the flow around the obstacle. On the contrary, we noticed that the superior shear layer is dissipated, which could damage the prediction of its mouvement. The corrected flow (see Figure 5.3) shows a small correction but already noticeable after 1/10 of the main frequency of detached eddies. An important point is that the extra CPU is less than 80%.

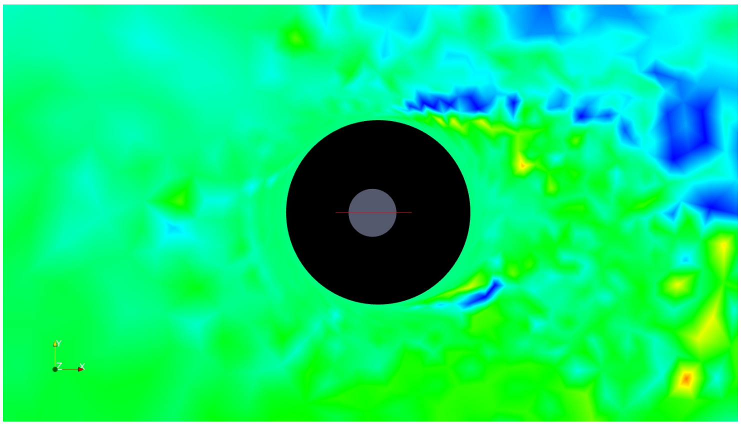


Figure 5.2: Numerical dissipation quantification: $(E_0(W_h) - E_1(W_h))$

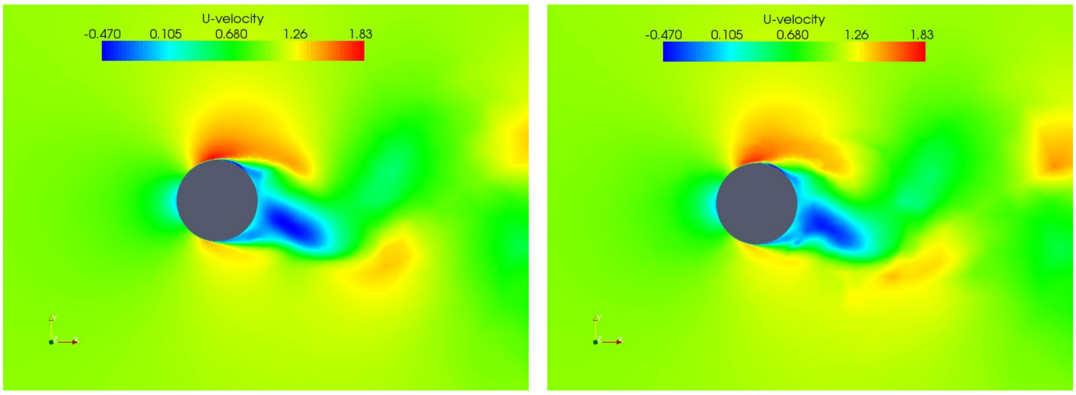


Figure 5.3: Instantaneuous velocity profile without correction (left) and respectively instantaneous velocity profile with numerical dissipation correction (right).

5.5 Conclusion

This section was devoted to the study of the use of an *a posteriori* estimate for evaluating a functional error and/or correcting a functional evaluation. We have clarified in which conditions computing a linearised error should be prefered to computing an adjoint. We have developed a platform for the simultaneous calculation of the flow with a corrected flow. A preliminary numerical example shows that this can be computed efficiently. This preliminary study has been exploited for the writing of the INRIA contribution in a new proposal for an European Project. The contribution will consist in proposing an accurate probabilistic model based on the error estimate.

Chapter 6

A priori variational error analysis

Contents

6.1 Summary

The lecturer of this chapter will be guided through an *a priori* finite-element error analysis where we will show how the approximation error is bounded by interpolation error, firstly for the usual Poisson equation, moving next to inviscid Euler flows and concluding with viscous flows. The principal interest of this error estimator based on interpolation error is for mesh adaptation problems for which dealing with interpolation error is the starting point.

6.2 Introduction

State of the art on mesh adaptation vs. error estimation: An important difficulty in mesh adaptation is the choice of the quantities that will prescribe the fineness and topology of the adapted mesh. Indeed, the approximation error for a PDE is a complex non-local function of mesh. The main ideas to solve this difficulty is to identify the error either as the solution of a continuous or the solution of a discrete linear error system, both with a right hand side, the local error, which depend on the local quality of the mesh. Two families of mesh adaptation methods can be distinguished. In a first set of works, a particular function of the dependant variable, the **sensor**, is chosen and its derivatives are considered as a good local truncation error indicator. In the case of Hessian-based adaptation, the proposed indicator is derived from the interpolation error in L^p for the sensor. The paramount interest of the Hessian-based approach for anisotropic adaptation is a very sound indication of the necessary stretching of the mesh. Another important feature is that it involves the research of an ideal mesh. However, in the case of finite-element approximations, the usual *a priori* error analysis shows that approximation error is smaller than the interpolation error only in the Sobolev space H^1, a different (stronger) norm than L^p .

In a second series of works, several authors propose a more accurate definition of the purpose of the adaptation, refered as *goal-oriented* method. It is to minimize the error on a real-valued functional. The error system of the PDE is taken into account through an adjoint state. Adjoint-based *a posteriori* error estimates were systematically derived from this specification. This allowed to compute an error level and to refine the regions where the local truncation error is larger than a tolerance ([Becker 1996a, Venditti 2002, Venditti 2003a]). The fact that effective *a posteriori* errors are controlled by the algorithm is an advantage of this kind of methods. These works are progressing towards anisotropic adaptation (see for example [Venditti 2003a, Apel 2007, Perotto 2008, Huang 2010]). One difficulty on this way is that corrections with respect to an existing mesh are managed instead of the specification of an ideal mesh.

This chapter is devoted to *a priori* models which combine goal-orientation and anisotropy. It is organized as following: we start with an *a priori* finite-element analysis applied to a simplified Poisson model, then the same analysis is proposed for unsteady Euler equations and the last section of this chapter details the *a priori* finite-element estimator applied to viscous Navier Stokes system.

6.3 A priori finite-element analysis

A priori estimates have been derived very earlier, in $H^1(\Omega)$ ("projection property"), and in $L^2(\Omega)$ (Aubin-Nitsche analysis, see for example [Nakao 1998]), but only by means of inequalities, and the leading term of the error is generally not exhibited (only bounds of it are proposed).

In this section, we try to go a little further than the standard *a priori* analysis. We concentrate on the usual Poisson problem, set in a polyhedral n-dimensional domain Ω for the sake of simplicity:

$$-\Delta u \;=\; f \;\; on \;\; \Omega \;\; ; \;\; u = 0 \;\; on \;\; \partial\Omega. \tag{6.1}$$

Its variational form writes:

$$a(u,v) \;=\; \int_\Omega \nabla u . \nabla v \,\mathrm{d}\mathbf{x} \;=\; < f \,,\, v > \qquad \forall\, v \in V \tag{6.2}$$

where V holds again for the Sobolev space

$$H_0^1(\Omega) = \{u \in L^2(\Omega), \nabla u \in (L^2(\Omega))^n, u|_{\partial\Omega} = 0\}.$$

Let $\mathcal{T}_h$ be a mesh of Ω made of simplices, and let V_h be the subspace of V of continuous functions that are $\mathcal{P}_1$ on each element of the mesh. The discrete variational problem is thus defined by:

$$a(u_h, v_h) \quad = \quad < \, f \, , \, v_h > \qquad \forall \, v_h \in V_h \tag{6.3}$$

Assuming that u is smooth enough, we can take its linear interpolation $\Pi_h u$ from vertices values. The approximation error in the sequel will be split into two components:

$$u_h \; - \; u \quad = \quad u_h \; - \; \Pi_h u \quad + \quad \Pi_h u \; - \; u \; . \tag{6.4}$$

We recognize in the second difference $\Pi_h u \; - \; u$ the *interpolation error*, and we shall refer to the first difference $u_h \; - \; \Pi_h u$ as the *implicit error*.

The implicit error is inside the discrete approximation space. Classically we can estimate it by writing that the finite-element solution is better than a direct projection of the continuous unknown. We try to proceed with equalities. We assume for simplicity that the right-hand side f is exactly integrated. The discrete system writes:

$$a(u_h, v_h) \quad = \quad (\, f, v_h) \; \forall \; v_h \in V_h$$

Let $\bar{\Pi}_h$ be a projection operator from V onto V_h. Since v is not necessarily smooth enough, it may be necessary to consider an interpolation operator like the Clement interpolation [Clément 1975]. For any v in V its projection $\bar{\Pi}_h v$ is in V_h, thus

$$a(u_h, \bar{\Pi}_h v) \quad = \quad (\, f, \bar{\Pi}_h v \;) \; \forall \; v \in V$$

then appears the continuous differential operator:

$$(\; (-\Delta) u_h \; , \; \bar{\Pi}_h v \;) \quad = \quad (\; f, \bar{\Pi}_h v \;) \; \forall \; v \in V$$

where the first product is the duality one between $H_0^1(\Omega)$ and $H^{-1}(\Omega)$. We also use the adjoint of the projector:

$$(\; \bar{\Pi}_h^*(-\Delta) u_h \; , \; v \;) \quad = \quad (\; \bar{\Pi}_h^* f \; , \; v \;) \; \forall \; v \in V$$

Thus

$$\bar{\Pi}_h^*(-\Delta) u_h \quad = \quad \bar{\Pi}_h^* f \;\; in \;\; V'$$

where by V' we denoted the dual space of V, which is $[H_0^1(\Omega)]'$. This allows the following error analysis:

$$\begin{aligned} \bar{\Pi}_h^*(-\Delta) u_h \; - \; \bar{\Pi}_h^*(-\Delta)\bar{\Pi}_h u \quad &= \quad \bar{\Pi}_h^* f \; - \; \bar{\Pi}_h^*(-\Delta)\bar{\Pi}_h u \;\; in \;\; V' \\ \bar{\Pi}_h^*(-\Delta)(u_h \; - \; \bar{\Pi}_h u) \quad &= \quad \bar{\Pi}_h^*(-\Delta)(u \; - \; \bar{\Pi}_h u) \;\; in \;\; V' \end{aligned} \tag{6.5}$$

where $\bar{\Pi}_h^*(-\Delta)$ can be inverted on adhoc spaces, as it is shown by the lemma below:

Lemma 6.3.1
[Courty 2005] The operator $\widetilde{\Pi_h^* A}$:

$$\begin{array}{ll} \widetilde{\Pi_h^* A} : & V_h \to \Pi_h^* A(V) \\ \widetilde{\Pi_h^* A} : & v_h \mapsto \Pi_h^* A v_h \ . \end{array}$$

is an isomorphism, where V_h is equipate with the Hilbert norm and $\Pi_h^* A(V)$ is equiped with the V' norm.□

Going back to our error estimator we conclude that:

$$u_h \ - \ \bar{\Pi}_h u \ = \ \left(\bar{\Pi}_h^*(-\Delta)\right)^{-1} \bar{\Pi}_h^*(-\Delta)(u \ - \ \bar{\Pi}_h u) \ ,$$

or, equivalently in variational form:

$$< \nabla(u_h - \bar{\Pi}_h u), \nabla\bar{\Pi}_h \varphi > \ = \ < \nabla(u - \bar{\Pi}_h u), \nabla\bar{\Pi}_h \varphi > \qquad (6.6)$$

For u and φ smooth, and by a density argument, the above calculation extends to the usual interpolation Π_h which will be considered in the sequel.
This is an *a priori* analysis since we express the error $u_h \ - \ \Pi_h u$ as a function of the unknown u.

Orientation: To fix our vocabulary, we observe that we have identified three terms in our error analysis:

- the interpolation or projection error $u \ - \ \Pi_h u$, it is a *local error* in the sense that a local change in the mesh would produce only local a change in this error,

- its complement to global approximation error is the *implicit error*

  $$E_h \ = \ u_h \ - \ \Pi_h u \ ,$$

  a nonlocal error, but the implicit error is the solution of a (discrete) elliptic system with as right-hand side the truncation error,

  $$< \nabla E_h, \nabla \Pi_h \varphi > \ = \ < \nabla(u - \Pi_h u), \nabla \Pi_h \varphi > \qquad (6.7)$$

- the *truncation error* $\Pi_h^*(-\Delta)(u \ - \ \Pi_h u)$ is a local error that is, in the present case, expressed in terms of the interpolation error.

In the sequel we go furthermore and develop this truncation error, firstly for the two dimensional case and then for the three dimensional one.

6.4 2D Truncation error analysis

The problem of the higher-order (second-order) convergence of the gradient of interpolation or approximation is refered as the recovery problem [Zienkiewicz 1992a, Zienkiewicz 1992b, Krizek 1998]. It can be solved for a regular enough mesh.

We concentrate in this section on analysing the right-hand side of Estimation (6.7), that is:

$$< \nabla(u - \Pi_h u), \nabla\Pi_h\varphi > \tag{6.8}$$

We restrict our analysis to the case of a family of isotropic meshes in the sense that they have a quasi-uniform aspect ratio equal to unity. A mesh of this family is represented by a scalar local mesh size $\Delta x = \Delta y = m(x, y)$.

Further, we are considering *h-families of meshes* for which the local mesh size $m_h(x, y)$ satisfies:

$$m_h(x, y) \;=\; h\; m(x, y),$$

where the normalized function $m(x, y)$ does not depend on h.
An important consequence is that, as far as the normalized metric is smooth enough we automatically assume that for a fine enough mesh, the variation of mesh size between two neiboring cells can be made as small as we wish.
In such a extremely regular mesh, the main part of an approximation error can show compensations between two neighboring elements. For a smooth function u, the approximate gradient $\nabla\Pi_h u$ can be of first order accuracy on a given equilateral triangle $T_+ = ABC, CA = CB$, while some of second order convergence can be obtained on the union of this triangle with the triangle T_- symmetric with respect to the basis AB (cf. Figure 6.1).

Decomposing the previous Estimate (6.7) into an integral on each element leads to:

$$\begin{aligned} < \nabla(u - \Pi_h u), \nabla\Pi_h\varphi > \;&=\; \int_\Omega \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi \right) \mathrm{d}\mathbf{x} \\ &=\; \sum_{T,\text{element}} \int_T \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi \right) \mathrm{d}\mathbf{x} \end{aligned}$$

where the sum Σ is taken over any element T of the mesh. Restricting to the integral over T for x-terms contribution and applying a Green formula we get:

$$\int_T \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi \right) \mathrm{d}\mathbf{x} \;=\;$$

$$\int_{\partial T} (u - \Pi_h u)(\frac{\partial}{\partial x}\Pi_h\varphi) \cdot n_x^T \mathrm{d}\sigma \;-\; \int_T \left((u - \Pi_h u) \left(\frac{\partial^2}{\partial x^2}\Pi_h\varphi \right) \right) \mathrm{d}\mathbf{x}$$

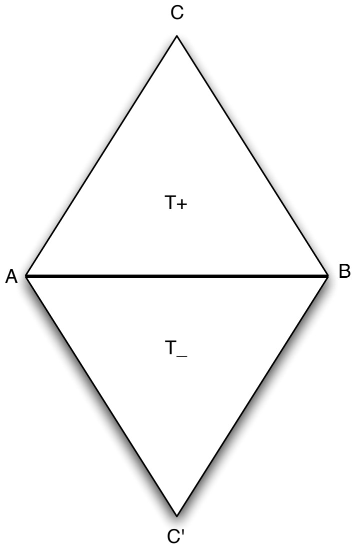


Figure 6.1: Superconvergent molecule for a vertical derivative

where by n_x^T we denoted the x component of the outward normal to the triangle T : $\mathbf{n}^T = (n_x^T, n_y^T)$. The same analysis holds for y terms. We observe that since $\Pi_h\varphi$ is $\mathcal{P}^1$ on T, the last integral in the right-hand side vanishes. Further, the derivative in the first integral is constant, hence:

$$\int_{\partial T}(u-\Pi_h u)(\frac{\partial}{\partial x}\Pi_h\varphi)\cdot n_x^T \mathrm{d}\sigma \;=\; \left(\frac{\partial}{\partial x}\Pi_h\varphi\right)|_T \int_{\partial T}(u-\Pi_h u)\cdot n_x^T \mathrm{d}\sigma.$$

Considering we are in the situation picturized in Figure 6.1, then the integral over ∂T applies to the two triangles T_+ and T_- with edge e as a common edge. We observe that the sum of integrals along e provided by the two triangles gives:

$$\left(\frac{\partial}{\partial x}\Pi_h\varphi\right)|_{T_+}\int_{\partial T_+\cap e}(u-\Pi_h u)\cdot n_x^{T_+}\mathrm{d}\sigma + \left(\frac{\partial}{\partial x}\Pi_h\varphi\right)|_{T_-}\int_{\partial T_-\cap e}(u-\Pi_h u)\cdot n_x^{T_-}\mathrm{d}\sigma$$
$$= (\frac{\partial}{\partial x}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial x}\Pi_h\varphi|_{T_-})\int_{\partial T_+\cap e}(u-\Pi_h u)\cdot n_x^{T_+}\mathrm{d}\sigma\ .$$

Remembering that $\Pi_h\varphi$ is continuous along any edge e, we can write:

$$\begin{aligned}(\frac{\partial}{\partial x}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial x}\Pi_h\varphi|_{T_-})\cdot n_x^{T_+}\int_{\partial T_+\cap e}(u-\Pi_h u)\mathrm{d}\sigma \;&=\\ (\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-})\int_{\partial T_+\cap e}(u-\Pi_h u)\mathrm{d}\sigma\ .\end{aligned} \tag{6.9}$$

In the case of a regular mesh as in Figure 6.1, the two terms of the difference $(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-})$ are derivatives evaluated at mid-altitudes:

Lemma 6.4.1
For a regular enough mesh, the term $\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right)$ *is consistent with a second normal derivative weighted by the inverse of mean altitude of the two triangles.*□

Proof: For the proof of this result we refer to Figure 6.1. Suppose we denote I the middle of segment $[AB]$ (also denoted edge e), then $\varphi(I) = (\varphi(A) + \varphi(B))/2$. Next, the difference $\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right)$ can be expressed by divided difference as:

$$\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-} = \frac{(y_I - y_{C'})\,(\varphi(C) - \varphi(I)) + (y_C - y_I)\,(\varphi(C') - \varphi(I))}{(y_C - y_I)(y_I - y_{C'})}.$$

By hypothesis the mesh is regular enough such that the two triangles (ABC) and (ABC') can be considered as equilateral or isosceles, thus we assume that: $y_C - y_I = y_I - y_{C'} = h$. Replacing with h where possible and after further computation we get for the right-hand side of the previous relation:

$$\frac{(y_I - y_{C'})\,(\varphi(C) - \varphi(I)) + (y_C - y_I)\,(\varphi(C') - \varphi(I))}{(y_C - y_I)(y_I - y_{C'})} = h \cdot \frac{\varphi(C) - 2\varphi(I) + \varphi(C')}{h^2},$$

where we recognize an estimation of the second order derivative from finite difference theory.
To conclude, the following estimate holds:

$$\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right) \approx h.\frac{\partial^2\varphi}{\partial y^2}(I)$$

In practice the jump for T_+ and T_- could differ, that is why we consider the mean jump $h = \frac{a_+ + a_-}{2}$, where by a_+ we denoted the altitude of T^+ triangle and respectively a_- the altitude of the T^- triangle.

□

Regarding y-terms contribution, the previous estimate holds too.

To synthetize, we have shown until here that:

$$\int_\Omega \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi\right) \mathrm{d}\mathbf{x} = \sum_{e,\text{edge}} [\nabla\Pi_h\varphi \cdot n]_e \int_e (u - \Pi_h u)_e \,\mathrm{d}e \qquad (6.10)$$

where according to the previous Lemma 6.4.1 the jump $[\nabla\Pi_h\varphi \cdot n]_e$ is identified as a second order derivative:

$$\int_\Omega \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi\right) \mathrm{d}\mathbf{x} \approx \sum_{e,\text{edge}} h \cdot \nabla^2\varphi \int_e (u - \Pi_h u)_e \,\mathrm{d}e \qquad (6.11)$$

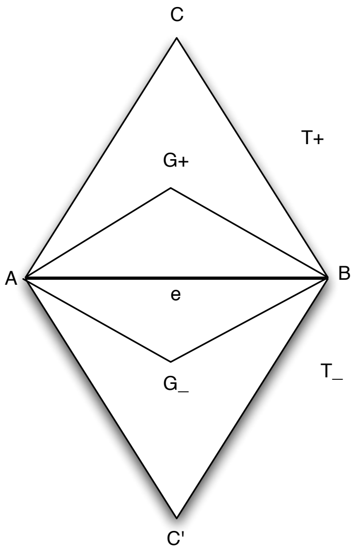


Figure 6.2: Diamont shape geometry : (AG_+BG_-)

Transformation of Estimate (6.11) into an integral on Ω. Once we have identify a second derivative, we can examine its weight. Indeed, it is multiplied by half the sum of the two altitudes of the triangles and integrated along the common edge (see proof of previous Lemma 6.4.1). Let us denote by G_+ (resp. G_-) the centroid of triangle T_+ (resp. T_-), a_+ (resp. a_-) the altitude of T_+ (resp T_-) orthogonal with respect to e. We shall refer in the sequel to the diamond-shaped surface D_e as the surface bounded by the segments joining either G_+ or G_- to an extremity of edge e (see Figure 6.2). The triangle formed by G_+ and e (or ABG_+), denoted here K_+, has an area of one third T_+'s area [1]:

$$|T_+| = \frac{1}{2} \cdot |e| \cdot |a_+| = 3 \cdot |K_+|.$$

Then, the sum of K^+ and K^- areas is equal to the area of the diamond-shaped surface D_e:

$$|D_e| = |K^+| + |K^-| = |e|\frac{|a_+| + |a_-|}{6} .$$

Let us approximate now the integral over edge e of the interpolation error $u - \Pi_h u$ from Estimate (6.11) with the integral over the diamond D_e for the same expression :

$$\frac{1}{|e|}\int_e (u - \Pi_h u)\ \mathrm{d}e \approx \frac{1}{|D_e|}\int_{D_e} (u - \Pi_h u)\ \mathrm{d}\mathbf{x}.$$

We observe that the union of diamond cells covers the whole computational domain

[1] $|\cdot|$ holds for volume, area or length of geometrical objects

Ω, *i.e.* $|\Omega| = \sum_{K\in\mathcal{H}} |K| = \sum_{D_e} |D_e|$. This allows to estimate the integral on Ω.

$$\int_\Omega \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi \right) \mathrm{d}\mathbf{x} \;\approx$$
$$\sum_{D_e} 3 \int_{D_e} (u - \Pi_h u)\nabla^2\varphi(\mathbf{x})\ \mathrm{d}\mathbf{x} = 3\int_\Omega (u - \Pi_h u)\nabla^2\varphi(\mathbf{x})\ \mathrm{d}\mathbf{x} \qquad (6.12)$$

A first estimate. If we try to write an estimate which does not depend too much on function φ, we can over-estimate the error as follows:

Lemma 6.4.2
We have the following bound:

$$|\int_\Omega (\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi) d\mathbf{x}| \;\preceq$$
$$3 \int_\Omega |\rho(H(\varphi)|\ |u - \Pi_h u| d\mathbf{x} \qquad (6.13)$$

where $A \preceq B$ holds for a majoration asymptotically valid, i.e. $A \leq B + O(A)$. Expression $|\rho(H(\varphi))|$ holds for the largest (in absolute value) eigenvalue of the Hessian $H(\varphi)$.□

6.5 3D Truncation error analysis

The previous steps from two dimensional analysis applies here too. Thus, we can skip the calculations and go directly to the main result and say:

Lemma 6.5.1
For a regular enough mesh, the term $(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-})$ is consistent with a second normal derivative weighted by the inverse of mean altitude of the two tetrahedra T_+ and T_-. □

To synthetize, as in two dimensional analysis, the following estimate holds:

$$\int_\Omega (\nabla(u - \Pi_h u)\nabla\Pi_h\varphi)\ \mathrm{d}\Omega \;\approx\; \sum_{f,\text{face}} [\nabla\Pi_h\varphi\cdot n]_f \int_f (u - \Pi_h u)_f\ \mathrm{d}\sigma\ . \qquad (6.14)$$

where the jump $\nabla\Pi_h\varphi \cdot n$ is estimated as a second order derivative, weighted by half the sum of corresponding altitudes of tetrahedras T_+ and T_-.

Transformation of Estimate (6.14) into an integral on Ω.

Let us denote by G_+ (resp. G_-) the centroid of tetrahedron T_+ (resp. T_-), a_+ (resp. a_-) the altitude of T_+ (resp T_-) with respect to f. The volume of the tetrahedron K^+ formed by G_+ and f is equal to:

$$|K^+| = \frac{1}{9}|f| \cdot |a_+|.$$

As for two dimensional case, we construct the diamond-shaped volume D_f bounded by the triangular plans joining either G_+ or G_- to a side of face f. Then, the volume of D_f is :

$$|D_f| = |K^+| + |K^-| = |f|\frac{|a_+| + |a_-|}{9}.$$

We shall approximate the integral over face f with the integral over the diamond D_f. Then again, the union of diamond cells covers the whole computational domain Ω, *i.e.* $|\Omega| = \sum |D_f|$. This allows to estimate the integral on Ω as:

$$\int_\Omega \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi \right) \mathrm{d}\mathbf{x} \approx \sum_{D_e} \frac{9}{2}\int_{D_e}(u - \Pi_h u)\nabla^2\varphi(\mathbf{x})\ \mathrm{d}\mathbf{x} = \frac{9}{2}\int_\Omega (u - \Pi_h u)\nabla^2\varphi(\mathbf{x})\ \mathrm{d}\mathbf{x} \tag{6.15}$$

A first $3D$ estimate. If we go furthermore and search for an estimate of (6.15) which does not depend too much on the function φ, we can over-estimate the error according to the following lemma:

Lemma 6.5.2
We have the following bound:

$$|\int_\Omega (\nabla(u - \Pi_h u)\nabla\Pi_h\varphi)d\Omega| \ \preceq\ \frac{9}{2}\ \int_\Omega |\rho(H(\varphi))|\ |u - \Pi_h u| d\Omega \tag{6.16}$$

where $A \preceq B$ holds for a majoration asymptotically valid, i.e. $A \leq B + O(A)$. Expression $|\rho(H(\varphi))|$ holds for the largest (in absolute value) eigenvalue of the Hessian $H(\varphi)$.□

Remark: In the case where $u - \Pi_h u$ does not vanish on the domain boundary denoted here Γ, then we get an extra term equivalent to:

$$\int_\Gamma (u - \Pi_h u)\nabla\varphi \cdot \mathbf{n}\ \mathrm{d}\sigma.$$

6.6 Inviscid flow of a compressible fluid

In the case of the Euler equations, we are not able to produce any theoretical theorem stating the convergence of the discrete solution to the continuous one. Also, we shall only discuss the case of smooth solutions in order to neglect higher order terms. The case of shocks can be adress with our methods as demonstrated in the thesis of Loseille [Loseille 2008]. Some explanations of this fact is given in [Loseille 2007], but a complete theory is missing. However we can formally apply the above strategy for estimating the right-hand side of the error equation.

We start from the finite-element formulation of Chapter 2 which writes:

$$\int_\Omega \phi_h \nabla.\mathcal{F}_h(W_h)d\Omega - \int_\Gamma \phi_h \bar{\mathcal{F}_h}^{out}(W_h)\cdot \mathbf{n}d\Gamma = \int_\Omega \phi_h \nabla.\mathcal{F}(W)d\Omega - \int_\Gamma \phi_h \bar{\mathcal{F}}^{out}(W)\cdot \mathbf{n}d\Gamma$$

where $\Gamma = \partial\Omega$ represents the boundary (solid and far-field) of the computational domain Ω.
We assume now that both W and ϕ are several times continuously differentiable. In order to estimate the error, we introduce on both sides the same expression with interpolations:

$$\underbrace{\int_\Omega \phi_h \nabla\cdot(\mathcal{F}_h(W_h) - \Pi_h\mathcal{F}(W))\mathrm{d}\Omega - \int_\Gamma \phi_h(\bar{\mathcal{F}_h}^{out}(W_h) - \Pi_h\bar{\mathcal{F}}^{out}(W))\cdot\mathbf{n}\ \mathrm{d}\Gamma}_{LHS}$$
$$= \underbrace{\int_\Omega \phi_h \nabla\cdot(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))\mathrm{d}\Omega - \int_\Gamma \phi_h(\bar{\mathcal{F}}^{out}(W) - \Pi_h\bar{\mathcal{F}}^{out}(W))\cdot\mathbf{n}\ \mathrm{d}\Gamma}_{RHS}.$$

The left-hand side (LHS) will be inverted and the right-hand side (RHS) will be expanded to get the error estimate.

6.6.1 Interpolation errors

We recall that $\Pi_h\mathcal{F}(W) = \mathcal{F}_h(W)$. The left-hand side writes:

$$(LHS, \phi_h) \ = \ \int_\Omega \phi_h\nabla\cdot(\mathcal{F}_h(W_h) - \mathcal{F}_h(W))\mathrm{d}\Omega - \int_\Gamma \phi_h(\bar{\mathcal{F}_h}^{out}(W_h) - \bar{\mathcal{F}}_h^{out}(W))\cdot\mathbf{n}\ \mathrm{d}\Gamma.$$

We linearize it as follows:

$$(LHS, \phi_h) \ \approx \ \int_\Omega \phi_h\nabla\cdot(\Pi_h\frac{\partial\mathcal{F}}{\partial W}(W_h - W))\mathrm{d}\Omega - \int_\Gamma \phi_h(\Pi_h\frac{\partial\bar{\mathcal{F}}^{out}}{\partial W}(W_h - W))\cdot\mathbf{n}\ \mathrm{d}\Gamma.$$

Where the derivatives $\frac{\partial\mathcal{F}}{\partial W}$ and $\frac{\partial\bar{\mathcal{F}}^{out}}{\partial W}$ are evaluated from vertex values of W. We denote this in short:

$$LHS \ = \ \mathcal{A}_h(W)(W_h - \Pi_h W).$$

We assume that the corresponding linearized operator, which is the Jacobian $\mathcal{A}_h(W)$ of the discretized Euler system is invertible. This means that the **implicit error** $W_h - \Pi_h W$ is obtained as the unique solution of:

$$W_h - \Pi_h W \ = \ (\mathcal{A}_h(W))^{-1} \ \ RHS\ .$$

In the right-hand side:

$$RHS \ = \ \int_\Omega \phi_h\nabla.(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))\mathrm{d}\Omega - \int_\Gamma \phi_h(\bar{\mathcal{F}}^{out}(W) - \Pi_h\bar{\mathcal{F}}^{out}(W))\cdot\mathbf{n}\ \mathrm{d}\Gamma$$

we recall that $\phi_h = \Pi_h\phi$ and next we add and substract a ϕ term:

$$RHS \;=\; RHS_1 + RHS_2$$

with:

$$\begin{aligned} RHS_1 \;=\; & \int_\Omega (\Pi_h\phi - \phi)\nabla\cdot(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))\mathrm{d}\Omega \\ & - \int_\Gamma (\Pi_h\phi - \phi)(\bar{\mathcal{F}}^{out}(W) - \Pi_h\bar{\mathcal{F}}^{out}(W))\cdot\mathbf{n}\;\mathrm{d}\Gamma\;. \end{aligned}$$

Assuming smoothness of ϕ and $\mathcal{F}(W)$, we deduce that on Ω, interpolation errors are of order two and their gradients are of order one, same on boundary, and RHS_1 is thus of order three:

$$RHS_1 \;\leq\; \mathrm{const}.h^3.$$

The second term writes:

$$RHS_2 \;=\; \int_\Omega \phi\nabla\cdot(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))\mathrm{d}\Omega - \int_\Gamma \phi(\bar{\mathcal{F}}^{out}(W) - \Pi_h\bar{\mathcal{F}}^{out}(W))\cdot\mathbf{n}\;\mathrm{d}\Gamma$$

and we transform it as follows:

$$\begin{aligned} RHS_2 \;=\; & -\int_\Omega (\nabla\phi)\cdot(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))\mathrm{d}\Omega \\ & + \int_\Gamma \phi(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))\cdot\mathbf{n}\;\mathrm{d}\Gamma \\ & - \int_\Gamma \phi(\bar{\mathcal{F}}^{out}(W) - \Pi_h\bar{\mathcal{F}}^{out}(W))\cdot\mathbf{n}\;\mathrm{d}\Gamma. \end{aligned}$$

The above estimates shows again the central role of the **interpolation error on internal and boundary fluxes** for the global approximation error.

Remark: In RHS_2 we can apply the same asymptotic extension as in the elliptic case studied in previous section. The expression of RHS_2 is in fact very good news. Indeed, due to the smoothness assumptions for ϕ and W, L^2 estimates for interpolation error on volume and on boundary apply, so that this term appears as a second-order one:

$$RHS_2 \;\leq\; \mathrm{const}.h^2.$$

Further, using the same techniques as in [Courty 2005], this term can be extended as follows:

$$RHS_2 \;=\; h^2\;(G(W,m),\phi) \;+\; R$$

where the last parenthesis is to be understood as a distribution one. The term R is of higher order:

$$R \;=\; o(h^2).\;\Box$$

6.6.2 Temporary conclusion

The above study shows that the implicit error $W_h - \Pi_h W$ is essentially a function of the interpolation error $W - \Pi_h W$. In the numerical applications we shall discard the boundary terms in order to simplify the mesh generation. In [Loseille 2009], it has been observed that this simplification does not reduce much the quality of the results.

6.7 Viscous flow of a compressible fluid

6.7.1 Notations

Let us refer now to the compressible Navier-Stokes system for a perfect gas. In conservative formulation it writes :

$$\frac{\partial W}{\partial t} + \nabla \cdot \mathcal{F} + \nabla \cdot \mathcal{V} = 0$$

where by $\mathcal{F}$ we refer to the Euler flux introduced previously in this thesis, namely in Chapter 2, and viscous flux is denoted $\mathcal{V}$.
We refer to Chapter 2 for details on the numerical resolution of this complex system. Nevertheless, we recall that we employ a mixed finite-element/volume discretisation for spatial resolution and viscous fluxes are discretised using P_1 finite-element method.
Since the Euler fluxes have been introduced and discussed in the previous section of the present chapter, we concentrate here on error analysis for viscous fluxes.

We describe in short the viscous fluxes as:

$$\mathcal{V} \quad = \quad [0, \sigma, -(\mathbf{q} - \mathbf{u}.\sigma)]^T,$$

where $\mathbf{u} = (u_1, u_2, u_3)$ is the velocity vector and the viscous stress tensor σ is defined as:

$$\sigma \quad = \quad \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) - \frac{2}{3}\mu \nabla.\mathbf{u}\mathbf{I},$$

with μ representing the constant viscosity.
The heat flux $\mathbf{q}$ is given by Fourier's law:

$$\mathbf{q} \quad = \quad -\lambda \nabla T$$

where λ is the heat conduction (assumed here to be constant), and T the temperature defined hereafter:

$$T = \frac{1}{c_v}\left(E - \frac{1}{2\rho}((\rho u_1)^2 + (\rho u_2)^2 + (\rho u_3)^2)\right),$$

with c_v also assumed to be constant.
The matrix expression of viscous fluxes $\mathcal{V}$ writes:

$$\begin{pmatrix} 0 & 0 & 0 \\ \sigma_{xx} & \sigma_{yx} & \sigma_{zx} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{zy} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \\ u.\sigma_{xx} + v.\sigma_{xy} + w.\sigma_{xz} + \lambda.\nabla T & u.\sigma_{yx} + v.\sigma_{yy} + w.\sigma_{yz} + \lambda.\nabla T & u.\sigma_{zx} + v.\sigma_{zy} + w.\sigma_{zz} + \lambda.\nabla T \end{pmatrix},$$

And viscous stress σ has the following general matrix expression:

$$\sigma = \mu \left[\underbrace{\begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix}}_{\nabla \mathbf{u}} + \underbrace{\begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{pmatrix}}_{\nabla \mathbf{u}^T} \right]$$

$$-\frac{2}{3}\mu \underbrace{\begin{pmatrix} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} & 0 & 0 \\ 0 & \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} & 0 \\ 0 & 0 & \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \end{pmatrix}}_{\nabla .\mathbf{uI}}$$

Suppose $\psi = (\psi_1, \psi_2, \psi_3, \psi_4, \psi_5)$ is the test function vector (or later on the adjoint vector). Following the *a priori* estimator of [Loseille 2010a] we are interested in the estimation of the different terms of the expression:

$$\mathcal{E} \;=\; (\Psi - \Psi_h, \Pi_h \psi)$$

where Ψ represents the continuous residual of the Navier-Stokes equations:

$$\Psi = W_t + \Phi^{\mathcal{F}} + \Phi^{\mathcal{V}}$$

and its discretisation Ψ_h is defined by:

$$\Psi_h = W_{h,t} + \Phi_h^{\mathcal{F}} + \Phi_h^{\mathcal{V}}.$$

We underline here that for the discretisation of viscous fluxes, we adopt the strategy of the elliptic study of Section 6.3 of the present chapter.
Viscous fluxes provide seven new terms to which we apply the finite-element error-analysis of Section 6.3, that is based on estimations of interpolation errors:

$$(\Phi^{\mathcal{V}}, \psi) = \int_\Omega \psi \nabla \cdot \mathcal{V} \, \mathrm{d}\Omega = \sum_{k=1}^{7} E_k \;.$$

The three first terms come from moment equations and depend only on $\psi_{234} = (\psi_2, \psi_3, \psi_4)^T$:

$$\begin{aligned} E_1 &= \int_\Omega \psi_{234} \nabla \cdot \mu \nabla \mathbf{u} \, \mathrm{d}\Omega \\ E_2 &= \int_\Omega \psi_{234} \nabla \cdot \mu (\nabla \mathbf{u})^T \mathrm{d}\Omega \\ E_3 &= -\frac{2}{3} \int_\Omega \psi_{234} \nabla \cdot \mu \nabla .\mathbf{uI} \, \mathrm{d}\Omega. \end{aligned}$$

The four last terms are derived from the energy equation:

$$\begin{aligned}
E_4 &= \int_\Omega \psi_5 \nabla \cdot \lambda \nabla T \mathrm{d}\Omega \\
E_5 &= \int_\Omega \psi_5 \nabla \cdot (\mathbf{u}.\mu \nabla \mathbf{u}) \,\mathrm{d}\Omega \\
E_6 &= \int_\Omega \psi_5 \nabla \cdot \left(\mathbf{u}.\mu (\nabla \mathbf{u})^T\right) \mathrm{d}\Omega \\
E_7 &= -\frac{2}{3} \int_\Omega \psi_5 \nabla \cdot (\mathbf{u}.\mu \nabla.\mathbf{u}\mathbf{I}) \,\mathrm{d}\Omega \,.
\end{aligned}$$

In the sequel, for each of these seven terms, we derive an error estimator following the *a priori* finite-element analysis of previous sections where we have shown how the implicit error is bounded by interpolation error weighted by some weights. The main idea of the resolution is to retrieve Relation (6.8) for which Lemma 6.5.2 can be applied.

6.7.2 Study of truncation error terms

As for the Euler flows case we do not consider in the sequel the boundary error terms.

Study of E_1, E_2 and E_3

Let us concentrate first on the three terms summation derived from the momentum equation:

$$\begin{array}{rclcl}
E_1 &=& \mu \int_\Omega \psi_{234} \nabla \cdot \nabla \mathbf{u} d\Omega &=& \mu \sum_{i=1}^3 \sum_{j=1}^3 \int_\Omega \psi_{i+1} \frac{\partial}{\partial x_j}\left(\frac{\partial}{\partial x_j} u_i\right) d\Omega, \\
E_2 &=& \mu \int_\Omega \psi_{234} \nabla \cdot (\nabla \mathbf{u})^T d\Omega &=& \mu \sum_{i=1}^3 \sum_{j=1}^3 \int_\Omega \psi_{i+1} \frac{\partial}{\partial x_j}\left(\frac{\partial}{\partial x_i} u_j\right) d\Omega, \\
E_3 &=& -\mu \frac{2}{3} \int_\Omega \psi_{234} \nabla \cdot (\nabla.\mathbf{u}\mathbf{I}) d\Omega &=& -\mu \frac{2}{3} \sum_{i=1}^3 \sum_{j=1}^3 \int_\Omega \psi_{i+1} \frac{\partial}{\partial x_i}\left(\frac{\partial}{\partial x_j} u_j\right) d\Omega.
\end{array}$$

We remark that E_2 and E_3 expressions can be directly additioned with an exchange of i and j derivatives. Then the summation of these terms writes:

$$E_1+E_2+E_3 = \mu \left(\sum_{i=1}^3 \sum_{j=1}^3 \int_\Omega \psi_{i+1} \frac{\partial}{\partial x_j}\left(\frac{\partial}{\partial x_j} u_i\right) d\Omega \;+\; \frac{1}{3} \sum_{i=1}^3 \sum_{j=1}^3 \int_\Omega \psi_{i+1} \frac{\partial}{\partial x_i}\left(\frac{\partial}{\partial x_j} u_j\right) d\Omega \right).$$

It is sufficient to consider e_{123} :

$$e_{123} = \int_\Omega \Pi_h \psi_{i+1} \frac{\partial^2 u_l}{\partial x_i \partial x_j} d\Omega.$$

The other terms are depicted identically.
We are interested in the error term:

$$\begin{aligned}
\delta e_{123} = e_{123} - e_{123}^h &= \int_\Omega \Pi_h \psi_{i+1} \frac{\partial^2 u_l}{\partial x_i \partial x_j} d\Omega - \int_\Omega \Pi_h \psi_{i+1} \frac{\partial^2 \Pi_h u_l}{\partial x_i \partial x_j} d\Omega \\
&= \int_\Omega \Pi_h \psi_{i+1} \frac{\partial^2}{\partial x_i \partial x_j} (u_l - \Pi_h u_l) d\Omega.
\end{aligned}$$

After a first integration by part and neglecting the boundary terms, the error term writes:

$$\delta e_{123} = -\int_\Omega \frac{\partial}{\partial x_i}\left(\Pi_h \psi_{i+1}\right)\frac{\partial}{\partial x_j}(u_l - \Pi_h u_l)\ d\Omega.$$

Now, according to Lemma 6.5.2 from the elliptic error analysis these volumic contribution is overestimated as:

$$\delta e_{123} \preceq \frac{9}{2}\int_\Omega \rho(H(\psi_{i+1}))|u_l - \Pi_h u_l|.$$

And finally, going back to our initial sommation the following *a priori* estimate holds for the first three terms of viscous flux contribution:

$$\delta E_1 + \delta E_2 + \delta E_3 \preceq \frac{9}{2}\mu\sum_{i=1}^{3}\left(3\ \rho(H(\psi_{i+1})) + \frac{1}{3}\sum_{j=1}^{3}\rho(H(\psi_{i+1}))\right)|u_i - \Pi_h u_i|.$$

We recall that 9/2 constant comes from the three dimensional estimator, and since the first term of the summation is independent on j, it is thus multiplied by 3.

Study of E_4

The first term from the energy equation is discussed now:

$$E_4 \ = \int_\Omega \psi_5 \nabla\cdot\lambda\nabla T \mathrm{d}\Omega.$$

We consider the following discretisation :

$$E_{4,h} \ = \int_\Omega \Pi_h\psi_5 \nabla\cdot\lambda\nabla\Pi_h T \mathrm{d}\Omega.$$

We focus on the error term:

$$\delta E_{4,h} \ = \int_\Omega \Pi_h\psi_5 \nabla\cdot\lambda\nabla T \mathrm{d}\Omega - \int_\Omega \Pi_h\psi_5 \nabla\cdot\lambda\nabla\Pi_h T \mathrm{d}\Omega.$$

After a first integration by parts we get:

$$\begin{aligned}\delta E_{4,h} &= \int_\Omega \Pi_h\psi_5\lambda\nabla\cdot\left(\nabla(T - \Pi_h T)\right)\mathrm{d}\Omega \\ &= \int_\Gamma \lambda\ \Pi_h\psi_5\nabla(T - \Pi_h T)\cdot\mathbf{n}\ \mathrm{d}\Gamma - \int_\Omega \lambda\nabla(\Pi_h\psi_5)\left(\nabla(T - \Pi_h T)\right)\mathrm{d}\Omega.\end{aligned}$$

The boundary terms contribution is neglected as already mentioned. Then, the volume integral is equivalent to Estimation (6.8) for which Lemma 6.5.2 can be applied.

We obtain thus the final estimate:

$$\delta E_{4,h} \preceq \frac{9}{2}\int_\Omega |\lambda|\rho(H(\Psi_5))\ |\Pi_h T - T|\mathrm{d}\Omega. \tag{6.17}$$

For the next three remaining terms, E_5, E_6 and E_7, because of their non-linearity, a slightly different algorithm of resolution is employed, by some mathematical artifice.

Study of E_5

We start from the following developpement:

$$E_5 = \mu \sum_{i=1}^{3}\sum_{j=1}^{3} \int_\Omega \psi_5 \frac{\partial}{\partial x_j}\left(u_i \frac{\partial u_i}{\partial x_j}\right) d\Omega.$$

We resume next to the integral formulation and analyse the error term:

$$\delta e_5 = e_5 - e_5^h = \int_\Omega \Pi_h \psi_5 \frac{\partial}{\partial x_j}\left(u_i \frac{\partial u_i}{\partial x_j}\right) d\Omega - \int_\Omega \Pi_h \psi_5 \frac{\partial}{\partial x_j}\left(\Pi_h u_i \frac{\partial \Pi_h u_i}{\partial x_j}\right) d\Omega.$$

Because of the non-linearity of this term, we cannot directly add the two integrals and perform an integration by parts for this summation. In order to obtain a relation equivalent to (6.8) we perform firstly an integration by part for each one of the two integrals and discard the boundary terms.

$$\delta e_5 = -\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h \psi_5)\left(u_i \frac{\partial u_i}{\partial x_j}\right) d\Omega + \int_\Omega \frac{\partial}{\partial x_j}(\Pi_h \psi_5)\left(\Pi_h u_i \frac{\partial \Pi_h u_i}{\partial x_j}\right) d\Omega =$$
$$\underbrace{-\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h \psi_5)\left(u_i \frac{\partial (u_i - \Pi_h u_i)}{\partial x_j}\right) d\Omega}_{I_1} + \underbrace{\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h \psi_5)\,(\Pi_h u_i - u_i)\,\frac{\partial \Pi_h u_i}{\partial x_j} d\Omega}_{I_2}.$$

Next, regarding integral I_1, after an integration by parts we have:

$$I_1 = -\int_\Omega \left(\frac{\partial^2}{\partial x_j^2}(\Pi_h \psi_5)\right) u_i\ (u_i - \Pi_h u_i) d\Omega +$$
$$\underbrace{\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h \psi_5)\frac{\partial u_i}{\partial x_j}\ (u_i - \Pi_h u_i) d\Omega}_{T_{11}} - \underbrace{\int_\Gamma u_i \left(\frac{\partial}{\partial x_j}(\Pi_h \psi_5)\right) \cdot n_j\ (u_i - \Pi_h u_i) d\Gamma}_{T_{12}}.$$

We recognize in T_{12} the hypothesis of Lemma 6.5.2. Thus:

$$T_{12} \preceq \frac{9}{2} \int_\Omega |u_i| \rho(H(\psi_5))\ |u_i - \Pi_h u_i|\ d\Omega$$

Regarding T_{11} terms, we observe they are (closed to) identical to I_2, but opposite signs, thus we discard them.

Finally, the total contribution of E_5 term writes then:

$$\delta E_5 \preceq \frac{27}{2}\mu \sum_{i=1}^{3} \int_\Omega |u_i| \rho(H(\psi_5))\ |u_i - \Pi_h u_i|\ d\Omega. \tag{6.18}$$

Study of E_6

In contrast with the previous term, the gradient of the velocity vector is transposed, thus the velocity components will be crossed.

Let us write:

$$E_6 = \mu \sum_{i=1}^{3}\sum_{i=1}^{3} \int_\Omega \psi_5 \frac{\partial}{\partial x_j}\left(u_i \frac{\partial u_j}{\partial x_i}\right) d\Omega.$$

We analyse the error term:

$$\delta e_6 = \int_\Omega \Pi_h \psi_5 \frac{\partial}{\partial x_j}\left(u_i \frac{\partial u_j}{\partial x_i}\right) d\Omega - \int_\Omega \Pi_h \psi_5 \frac{\partial}{\partial x_j}\left(\Pi_h u_i \frac{\partial \Pi_h u_j}{\partial x_i}\right) d\Omega$$

After a first integration by parts applied on both integrals we obtain:

$$\begin{aligned}\delta e_6 &= -\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\left(u_i\frac{\partial u_j}{\partial x_i}\right) d\Omega + \int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\left(\Pi_h u_i \frac{\partial \Pi_h u_j}{\partial x_i}\right) d\Omega \\ &= \underbrace{-\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\left(u_i\frac{\partial (u_j - \Pi_h u_j)}{\partial x_i}\right) d\Omega}_{I_1} + \underbrace{\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\ (\Pi_h u_i - u_i)\ \frac{\partial \Pi_h u_j}{\partial x_i} d\Omega}_{I_2}\,.\end{aligned}$$

As for the previous term, a second integration by parts is applied with the assumption that the boundary terms can be neglected:

$$I_1 = \underbrace{\int_\Omega \frac{\partial \Pi_h \psi_5}{\partial x_j}\ \frac{\partial u_i}{\partial x_i}\,(u_j - \Pi_h u_j)\, d\Omega}_{T_{11}} - \underbrace{\int_\Gamma u_i \left(\frac{\partial \Pi_h\psi_5}{\partial x_j}\right)\cdot n_j\ (u_j - \Pi_h u_j)\ d\Gamma.}_{T_{12}}$$

We recognize in T_{12} the estimation of Lemma 6.5.2, that is:

$$T_{12} \preceq \frac{9}{2}\mu\ \int_\Omega |u_i|\ \rho(H(\psi_5))\ |u_j - \Pi_h u_j|\ d\Omega.$$

For the T_{11} and I_2 after further calculation we obtain the interpolation error on velocity vector weighted by a vector with, as components, cross-products of gradient of the velocity with gradient of ψ_5 . Suppose we denote this weight Vec, we have:

$$\delta E_6 \preceq \frac{9}{2}\mu\ \sum_{i=1}^{3}\sum_{j=1}^{3}\int_\Omega |u_i|\ \rho(H(\psi_5))|u_j - \Pi_h u_j|\ d\Omega + \mu \int_\Omega \text{Vec}\ |\mathbf{u} - \Pi_h \mathbf{u}|\ d\Omega.$$

with (for 3D problems):

$$\text{Vec} = \begin{pmatrix} (\nabla w \times \nabla\psi_5)_y - (\nabla v \times \nabla \psi_5)_z \\ (\nabla u \times \nabla\psi_5)_z - (\nabla w \times \nabla \psi_5)_x \\ (\nabla v \times \nabla\psi_5)_x - (\nabla u \times \nabla \psi_5)_y \end{pmatrix}$$

and respectively $\text{Vec} = (-\nabla v \times \nabla\psi_5, \nabla u \times \nabla \psi_5)^T$ for two dimensional case.

To synthetize, the following estimation holds:

$$\delta E_6 \preceq \mu \sum_{i=1}^{3} \int_\Omega \left(\sum_{j=1}^{3}\ (\frac{9}{2}|u_j|\ \rho(H(\psi_5)))\ +\ \text{Vec}[i] \right)\ |u_i - \Pi_h u_i| \qquad (6.19)$$

Study of E_7

The previous remark regarding the crossing of terms holds for the 7th term too, because of multiplication with indentity matrix.
This terms writes:

$$E_7 \;=\; -\frac{2}{3}\mu \sum_{k=1}^{3}\sum_{i=1}^{3}\sum_{j=1}^{3} \int_\Omega \psi_5 \frac{\partial}{\partial x_k}\left(u_i\left(\frac{\partial u_j}{\partial x_j}\right)\right) d\Omega$$

And the error term to be analysed can be restricted to:

$$\delta e_7 = \int_\Omega \Pi_h\psi_5 \frac{\partial}{\partial x_k}\left(u_i\left(\frac{\partial u_j}{\partial x_j}\right)\right) d\Omega \;-\; \int_\Omega \Pi_h\psi_5 \frac{\partial}{\partial x_k}\left(\Pi_h u_i\left(\frac{\partial \Pi_h u_j}{\partial x_j}\right)\right) d\Omega$$

After a first integration by parts the previous error term writes:

$$\begin{aligned}\delta e_7 \;=\; & -\int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\left(u_i\frac{\partial u_j}{\partial x_j} d\Omega + \int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\left(\Pi_h u_i\left(\frac{\partial \Pi_h u_j}{\partial x_j}\right)\right) d\Omega = \\ & \underbrace{-\int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\left(u_i\frac{\partial (u_j - \Pi_h u_j)}{\partial x_j}\right) d\Omega}_{I_1} + \underbrace{\int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\,(\Pi_h u_i - u_i)\left(\frac{\partial \Pi_h u_j}{\partial x_j}\right) d\Omega}_{I_2}.\end{aligned}$$

Furthermore, we apply a second integration by parts:

$$\delta e_7 \;=\; \underbrace{\int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\frac{\partial u_i}{\partial x_j}\,(u_j - \Pi_h u_j)\, d\Omega}_{T_{11}} \underbrace{-\int_\Gamma \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\; u_i\cdot n_i\;(u_j\Pi_h u_j) d\Gamma}_{T_{12}} + I_2.$$

We recognize in T_{12} the estimation from Lemma 6.5.2, thus the following estimation holds:

$$T_{12} \preceq \frac{9}{2}\int_\Omega |u_i|\;\rho(H(\psi_5))\;|u_j - \Pi_h u_j| d\Omega.$$

Regarding the integral I_2 the following estimation holds:

$$I_2 \approx \int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\left(\frac{\partial u_j}{\partial x_j}\right)\;\;(\Pi_h u_i - u_i)\, d\Omega$$

After summation we retrieve for this term the weight vector Vec from E_5, that is, for two dimensional case:

$$T_{11} + I_2 \;=\; -\,\mathrm{Vec}\;(\mathbf{u} - \Pi_h\mathbf{u}) \;+\; (v_x\psi_{5,x} - v_y\psi_{5,y}\;;\; u_y\psi_{5,y} - u_x\psi_{5,x})^T\;(\mathbf{u} - \Pi_h\mathbf{u})$$

Thus, the total contribution of E_7 term writes:

$$\delta E_7 \preceq -\frac{2}{3}\mu \sum_{k=1}^{3}\sum_{i=1}^{3}\sum_{j=1}^{3}\left[\frac{9}{2}\int_\Omega |u_i|\;\rho(H(\psi_5))\;|u_j - \Pi_h u_j| d\Omega + T_{11} + I_2\right]. \qquad (6.20)$$

6.7.3 Conclusion

This estimate is of paramount importance for one of the goals of this work, which is anisotropic mesh adaptation for functional outputs where interpolation errors needs to be managed. For the description of the mesh adaptation problem for viscous flows we refer to Chapter 9.

Chapter 7

Metric-based mesh representation

Contents

7.1 Summary

This chapter is a reminder of the main ingredients needed by an anisotropic metric-based mesh generator. The motivation for the choice of a mesher dealing with a Riemannian metric is presented in the introduction. We start next with the notion of Riemannian metric and reviewing the main operations on metrics. We evoke the notion of unit element and unit mesh, basic elements for our mesh generation process.

7.2 Introduction

One of the continuous desire of CFD researchers and engineers is a more accurate capture of the flow features and properties, an essential element for computational certification. In some cases, CFD solutions are computed with an unnecessarily large number of grid points (and associated high cost) to ensure that the solutions are computed to within a required accuracy. In most cases, the expected accuracy is not obtained due to insufficient resolution in some part of the computational domain. One of the powerful methods for increasing the accuracy and reducing the computational cost is the mesh adaptation, whose purpose is to control the accuracy of the numerical solution by changing the discretization of the computational domain according to mesh size and mesh directions constraints.

Traditionally, research has been concentrated on isotropic mesh adaptation where mesh elements are adjusted only in size according to an error estimate or indicator while their shape is kept close to being equilateral. However, isotropic meshes often tend to use too many elements in regions of large solution error. This is especially true when problems have an anisotropic feature which makes the solution changes more significantly in one direction than in the others. Full benefits of mesh adaptation can only be taken by simultaneously adjusting the size, shape, and orientation of mesh elements according to the behavior of the physical solution. This often results in an anisotropic mesh, a mesh having elements of large aspect ratio. This technique allows (i) to automatically capture the anisotropy of the phenomena, (ii) to subtantially reduce the number of degrees of freedom, thus impacting favorably the CPU time, , and (iii) to access to high order asymptotic convergence.

Many works in the literature proposed the mesh deformation or equivalently mesh coordinates as the solution of a particular system coupled with the discrete Partial Differential Equation (PDE) under study. More precisely, the adapted mesh is chosen among a set of deformed meshes derived from a starting reference mesh. See for example [Cabello 1991, Palmerio 1996]. In this case the mesh topology is prescribed by the user and may be not adequate for the adaptation.

Conversely, in the case where the user does not wish to fix the topology, but instead, wants the algorithm to find it, then the definition of a system the solution of which is the adapted mesh is much more difficult. Starting from a given mesh and locally refining it is a far from an optimal strategy. Let us discard that option and look for an optimal mesh. First we can observe that this is an ill-posed problem since two meshes can have very different topologies and give about the same local accuracy. Secondly, it is difficult to find an optimal mesh if we have to investigate inside a set of meshes described by integers and booleans.

These remarks have motivated researchers to represent meshes by continuous functions. See for example [Babuska 2001]. These functions can be for example the (scalar) local mesh density over the computational domain. From its knowledge, it is possible to derive an upper bound for the local truncation error. But this upper bound does not give a perfect idea of the local error if local stretching effects are not taken into account.

Two decades ago, several publications, see for example [Borouchaki 1996], [Frey 1999] and [Habashi 2000], propose to model the local stretching by means of a non-scalar field, the **metric**. An adapted metric is specified by an argument of equidistribution of an interpolation error related to the Partial Differential Equation solution.

The purpose of the present chapter is to present the adaptation method employed by our team and widely used in the next two chapters. The main feature is to describe the mesh in a general continuous setting, refered in the sequel as *metric*, or *continuous mesh*. This is important since the main result of this thesis is the demonstration of how the errors committed on Euler and Navier-Stokes systems are well represented with this concept of metric.

This chapter is organized as follows: in Section 7.4 we present the mesh generation process through the concept of unit mesh, then in Section 7.5 the operations on metrics are introduced, and we end this chapter with explanations on numerical computation of some geometrical quantities.

7.3 Riemannian metric space

Euclidean geometry it is most certainly the most used space nowadays for geometrical constructions. We can say that Euclidean geometry is the study of flat space, in the sense that all the geometrical objects can be represented on a flat piece of paper. Between every pair of points, there is a unique line segment which is the shortest curve between those two points.

We will introduce in the sequel a different geometrical space where, in a usual language we can say that the drawing is performed on a curved piece of paper.

In differential geometry, a Riemannian manifold or Riemannian space $(M, \mathcal{M})$ is a manifold possessing a metric tensor $\mathcal{M}$ varying smoothly from point to point. For a complete Riemannian manifold, the metric is defined as the length of the shortest curve (geodesic) between two points x and y.

For our own purpose of mesh adaptation, we work with a simpler object called **Riemannian metric space** defined by $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. The common notions of length, volume and angle in this Riemannian metric space are defined next. Fortunately, these notions can be easily derived in the context of meshing because we are not interested in evaluating these quantities on the Riemannian manifold. Indeed, as regards edge length computation, we do not want to compute the distance between two points which requires to find the shortest path on the curved manifold between these two points and to compute the length of the geodesic, but to compute the length of the path between these two points defined by the straight line parameterization, *i.e.,* the segment representing the edge between these two points in the parametrization space which is our computational domain. To take into account the variation of the metric along the edge, the edge length is evaluated with an integral formula.

Definition 7.3.1
In Riemannian metric space $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$*, the* **length** *of edge* **ab** *is computed using the straight line parameterization in domain* Ω $\gamma(t) = \mathbf{a}+t\,\mathbf{ab}$*, where* $t \in [0,1]$*:*

$$\ell_{\mathcal{M}}(\mathbf{ab}) = \int_0^1 \|\gamma'(t)\|_{\mathcal{M}}\,\mathrm{d}t = \int_0^1 \sqrt{{}^t\mathbf{ab}\;\mathcal{M}(\mathbf{a}+t\,\mathbf{ab})\;\mathbf{ab}}\,\mathrm{d}t. \tag{7.1}$$

Let us introduce next the notions of volume and angle in Riemannian metric spaces.

Definition 7.3.2
Given a bounded subset K of Ω, the **volume** *of K computed with respect to Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is:*

$$|K|_{\mathcal{M}} = \int_K \sqrt{\det \mathcal{M}(\mathbf{x})}\,\mathrm{d}\mathbf{x}\,. \tag{7.2}$$

For a Riemannian manifold, the angle between two curves is the angle between their tangent vectors at their point of intersection in the tangent plane. In our particular context, edge and tangent vector coincide, hence, we have:

Definition 7.3.3
The **angle** *between two edges* **pq** *and* **pr** *of Ω in Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is defined by the unique real-value $\theta \in [0,\pi]$ verifying:*

$$\cos(\theta) = \frac{\langle \mathbf{pq}, \mathbf{pr}\rangle_{\mathcal{M}(\mathbf{p})}}{\|\mathbf{pq}\|_{\mathcal{M}(\mathbf{p})}\|\mathbf{pr}\|_{\mathcal{M}(\mathbf{p})}}\,. \tag{7.3}$$

In three dimensions, the computation of a dihedral angle in a Riemannian metric space requires to use an integral formula as this angle varies depending on the position on the edge sharing both faces.

Definition 7.3.4
The **dihedral angle** *between two faces* [**pqr**] *and* [**sqp**] *of Ω sharing common edge* **pq** *in Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is defined by the real-value $\theta \in [0,\pi]$ verifying:*

$$\cos(\theta) = \int_0^1 \frac{\langle \mathbf{n_{pqr}}, \mathbf{n_{sqp}}\rangle_{\mathcal{M}(\mathbf{p}+t\mathbf{pq})}}{\|\mathbf{n_{pqr}}\|_{\mathcal{M}(\mathbf{p}+t\mathbf{pq})}\|\mathbf{n_{sqp}}\|_{\mathcal{M}(\mathbf{p}+t\mathbf{pq})}}\,\mathrm{d}t\,, \tag{7.4}$$

where $\mathbf{p} + t\,\mathbf{pq}$ with $t \in [0,1]$ is the straight line parameterization of edge **pq** *in domain Ω and, $\mathbf{n_{pqr}}$ and $\mathbf{n_{sqp}}$ are the oriented normals to* [**pqr**] *and* [**sqp**], *respectively.*

7.4 Generation of adapted anisotropic meshes

The inquiries of a mesher to generate anisotropic meshes can be the prescription of a size and orientation of elements, for each point of the domain of computation. As the notions of length, volume and angle has been defined we are now able to generate meshes in the Riemannian metric space. These informations will be transmitted to the mesher which will try to best fit these demands. The main idea of metric-based mesh adaptation, as introduced in [George 1991], is to generate **a unit mesh** in a prescribed Riemannian metric space.

In this section, we first introduce the definition of unit element and then the notion of unit mesh.

7.4.0.1 Unit element

Definition 7.4.1
An element K is **unit** *with respect to a metric $\mathcal{M}$ if the length of all its edges is unit in the metric $\mathcal{M}$.*

For instance, a tetrahedron K defined by its list of edges $(\mathbf{e_i})_{i=1..6}$ is unit with respect to $\mathcal{M}$ if:

$$\forall i = 1, ..., 6, \;\; \ell_{\mathcal{M}}(\mathbf{e_i}) = 1\,,$$

and its volume is: $|K|_{\mathcal{M}} = \dfrac{\sqrt{2}}{12}$. Figure 7.1 gives two examples of unit elements for two different metric tensors.

Let $\mathcal{M}$ be a metric tensor, there exists a non-empty infinite set of unit elements with respect to $\mathcal{M}$, as illustrated in Figure 7.2. Conversely, given an element K such that $|K|_{\mathcal{I}_n} \neq 0$, there is a unique metric tensor $\mathcal{M}$ for which element K is unit with respect to $\mathcal{M}$.

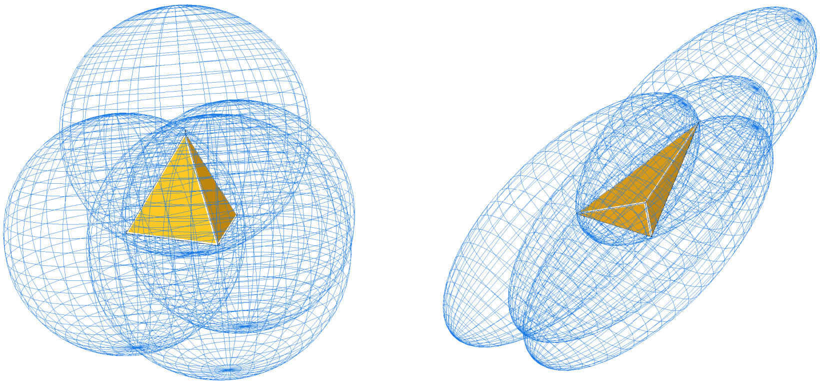


Figure 7.1: 3D examples of a unit element with respect to $\mathcal{I}_3$ (left) and to an anisotropic metric tensor $\mathcal{M}$ (right). In each case, the unit ball of the metric is drawn at each vertex of the unit element.

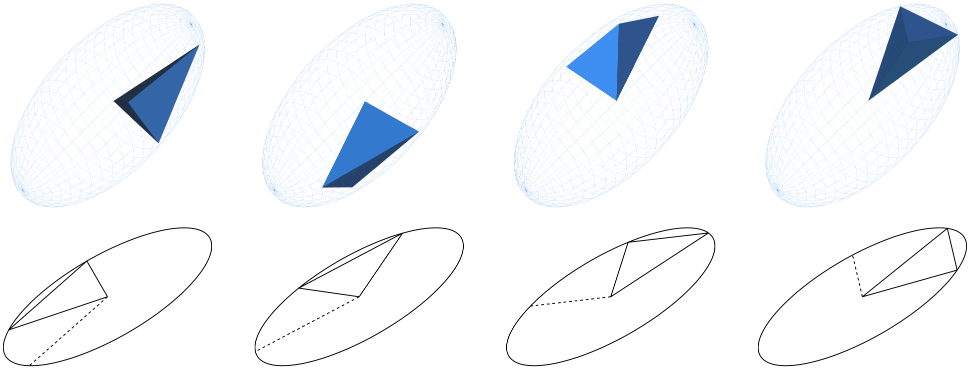


Figure 7.2: Several unit elements with respect to a metric tensor in 2D and 3D.

7.4.0.2 Unit mesh

By unit mesh we understand a mesh possessing only unit elements with respect to a given Riemannian metric space. Of course that generating such meshes is not always guaranteed.

Therefore, the notion of unit mesh has to be relaxed:

Definition 7.4.2
A discrete mesh $\mathcal{H}$ of a domain $\Omega \subset \mathbb{R}^n$ is **unit** *for Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ if all its elements are quasi-unit.*

By quasi-unit elements we mean that the length of edges $l_{\mathcal{M}}$ is contained into a so called admissible length interval of the form $[\frac{1}{\alpha}, \alpha]$ with $\alpha > 0$, see [Frey 2001]. Usually we consider $\alpha = \sqrt{2}$. Therefore, a tetrahedron K defined by its list of edges $(\mathbf{e_i})_{i=1...6}$ is said to be quasi-unit if $\forall i \in [1, 6]$, $\ell_{\mathcal{M}}(\mathbf{e_i}) \in [\frac{1}{\sqrt{2}}, \sqrt{2}]$. The question of existence of such meshes has been adressed by means of the space filling tetrahedra (see [Sommerville 1923] and [Goldberg 1974]).

This contrained on the edges length can leed to the generation of quasi-unit elements with a null volume. Thus, controling only the edges may not be sufficient. To avoid this kind of constructions we add a constraint called *quality of an element*. The quality is a constrained on the volume of an element K defined as:

$$Q_{\mathcal{M}}(K) = \frac{36}{3^{\frac{1}{3}}} \frac{|K|_{\mathcal{M}}^{\frac{2}{3}}}{\sum_{i=1}^{6} \ell_{\mathcal{M}}^2(\mathbf{e_i})} \in [0, 1]. \tag{7.5}$$

For the perfect regular tetrahedron, whatever its edges length, the quality function is equal to 1. For a null volume tetrahedron, $Q_{\mathcal{M}}$ is 0. Notice that $Q_{\mathcal{M}}$ only quantifies the gap to the regular tetrahedron shape.

The following definition of quasi-unit element, which is also practically used by mesh generators, can thus be formulated:

Definition 7.4.3
A tetrahedron K defined by its list of edges $(\mathbf{e_i})_{i=1...6}$ is said **quasi-unit** *for $\mathcal{M}$ if*

$$\forall i \in [1, 6], \quad \ell_{\mathcal{M}}(\mathbf{e_i}) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right] \quad \textit{and} \quad Q_{\mathcal{M}}(K) \in [\alpha, 1] \quad \textit{with} \quad \alpha > 0\,.$$

In our case, $\alpha = 0.8$ is an acceptable value.

7.4.0.3 Generating an adapted mesh

We have specified how one can generate a unit mesh of a domain Ω with respect to metric $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ and how it can be analysed in terms of size (legth of the edges) and shape (quality of the elements). Let us illustrate now with a significant example how this uniform and unit mesh in the Riemannian metric space is visualised as an adapted and anisotropic one in the Euclidean space.

Example 7.1
Figure 7.3 shows in background the France map in the meter metric and in foreground the France map in the TGV time metric, i.e., *the time spent by the TGV to go from one place to another. The left picture shows how a uniform cartesian grid on the France map in the meter metric is deformed after being mapped onto the France map in the TGV time metric. Exactly the same idea is used in the generation of adapted anisotropic meshes.*

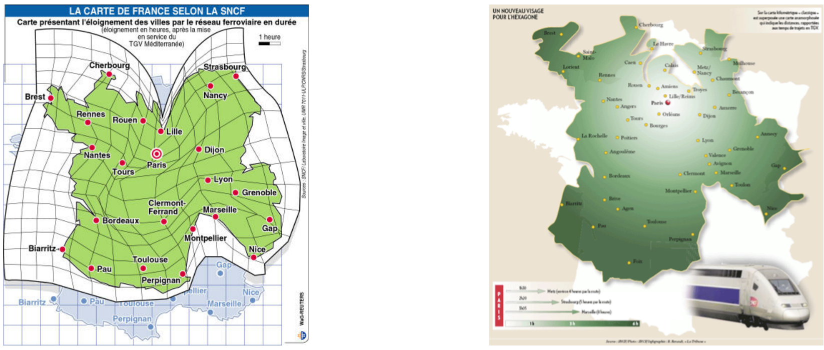

LA CARTE DE FRANCE SELON LA SNCF
Carte présentant l'éloignement des villes par le réseau ferroviaire en durée
(éloignement en heures, après la mise en service du TGV Méditerranée)
1 heure
Cherbourg
Strasbourg
Brest
Nancy
Rennes
Rouen
Lille
Nantes
Paris
Dijon
Tours
Lyon
Grenoble
Clermont-Ferrand
Marseille
Bordeaux
Gap
Montpellier
Biarritz
Nice
Toulouse
Pau
Perpignan
Sources : SNCF/ Laboratoire Image et ville, UMR 7011-ULP/CNRS/Strasbourg
WaG-REUTERS
UN NOUVEAU VISAGE POUR L'HEXAGONE

Figure 7.3: Two pictures representing the France map in the TGV time metric, *i.e.*, the time spent by the TGV to go from one place to another, as compared to the France in the meter metric.

Let us give a few words about the size prescription by a given Riemannian metric field. We have seen that the size of the unit ball associated with metric tensor $\mathcal{M}$ in the i$^{\text{th}}$ eigenvector direction is given by $h_i = \lambda_i^{-\frac{1}{2}}$. We then deduce that the prescribed size in direction $\mathbf{e}$ is

$$h_{\mathcal{M}}(\mathbf{e}) = \frac{\|\mathbf{e}\|_2}{\ell_{\mathcal{M}}(\mathbf{e})} \,.$$

In the case of an Euclidean metric space, the previous relation simply reads

$$h_{\mathcal{M}}(\mathbf{e}) = \frac{\|\mathbf{e}\|_2}{\sqrt{{}^t\mathbf{e}\mathcal{M}\mathbf{e}}} \,. \tag{7.6}$$

7.5 Operations on metrics

The main advantage when working with metric spaces is the well-posedness of operations on metric tensors, among which the metric intersection and the metric interpolation. These operations have a straightforward geometric interpretation when considering the ellipsoid associated with a metric.

7.5.1 Metric Intersection

When several metrics are specified at a point of the domain, all these metric tensors must be reduced to a single one due to mesh generation concerns. The **metric**

intersection consists in choosing in any direction the most restrictive size constraint imposed by this set of metrics.

Formally speaking, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two metric tensors given at a point. The metric tensor $\mathcal{M}_{1\cap 2}$ corresponding to the intersection of $\mathcal{M}_1$ and $\mathcal{M}_2$ is the one prescribing the largest possible size under the constraint that the size in each direction is always smaller than the sizes prescribed by $\mathcal{M}_1$ and $\mathcal{M}_2$. Let us give a geometric interpretation of this operator. Metric tensors are geometrically represented by an ellipse in 2D and an ellipsoid in 3D. But the intersection between two metrics is not directly the intersection between two ellipsoids as their geometric intersection is not an ellipsoid. Therefore, we seek for the largest ellipsoid representing $\mathcal{M}_{1\cap 2}$ included in the geometric intersection of the ellipsoids associated with $\mathcal{M}_1$ and $\mathcal{M}_2$, cf. Figure 7.4, left. The ellipsoid (metric) verifying this property is obtained by using the simultaneous reduction of two metrics.

Simultaneous reduction. The simultaneous reduction enables to find a common basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ such that $\mathcal{M}_1$ and $\mathcal{M}_2$ are congruent to a diagonal matrix in this basis, and then to deduce the intersected metric. To do so, the matrix $\mathcal{N} = \mathcal{M}_1^{-1}\mathcal{M}_2$ is introduced. $\mathcal{N}$ is diagonalizable with real-eigenvalues. The normalized eigenvectors of $\mathcal{N}$ denoted by $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ constitute a common diagonalization basis for $\mathcal{M}_1$ and $\mathcal{M}_2$. The entries of the diagonal matrices, that are associated with the metrics $\mathcal{M}_1$ and $\mathcal{M}_2$ in this basis, are obtained with the Rayleigh formula[1]:

$$\lambda_i = {}^t\mathbf{e}_i \mathcal{M}_1 \mathbf{e}_i \quad \text{and} \quad \mu_i = {}^t\mathbf{e}_i \mathcal{M}_2 \mathbf{e}_i\,, \quad \text{for} \quad i = 1\ldots 3\,.$$

Let $\mathcal{P} = (\mathbf{e}_1\, \mathbf{e}_2\, \mathbf{e}_3)$ be the matrix the columns of which are the eigenvectors $\{\mathbf{e}_i\}_{i=1\ldots 3}$ of $\mathcal{N}$. $\mathcal{P}$ is invertible as $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ is a basis of $\mathbb{R}^3$. We have:

$$\mathcal{M}_1 = {}^t\mathcal{P}^{-1} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \mathcal{P}^{-1} \quad \text{and} \quad \mathcal{M}_2 = {}^t\mathcal{P}^{-1} \begin{pmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{pmatrix} \mathcal{P}^{-1}\,.$$

Computing the metric intersection. The resulting intersected metric $\mathcal{M}_{1\cap 2}$ is then analytically given by:

$$\mathcal{M}_{1\cap 2} = \mathcal{M}_1 \cap \mathcal{M}_2 = {}^t\mathcal{P}^{-1} \begin{pmatrix} \max(\lambda_1, \mu_1) & 0 & 0 \\ 0 & \max(\lambda_2, \mu_2) & 0 \\ 0 & 0 & \max(\lambda_3, \mu_3) \end{pmatrix} \mathcal{P}^{-1}\,.$$

The ellipsoid associated with $\mathcal{M}_{1\cap 2}$ is the largest ellipsoid included in the geometric intersection region of the ellipsoids associated with $\mathcal{M}_1$ and $\mathcal{M}_2$, the proof is given in [Alauzet 2003a].

Numerically, to compute $\mathcal{M}_{1\cap 2}$, the real-eigenvalues of $\mathcal{N}$ are first evaluated with a Newton algorithm. Then, the eigenvectors of $\mathcal{N}$, which define $\mathcal{P}$, are computed using the algebra notions of image and kernel spaces.

[1] λ_i and μ_i are not the eigenvalues of $\mathcal{M}_1$ and $\mathcal{M}_2$. They are spectral values associated with basis $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$.

Remark 1

The intersection operation is not commutative. Consequently, when more than two metrics are intersected, the result depends on the order of intersection. In this case, the resulting intersected metric is not anymore optimal. If, we seek for the largest ellipsoid included in the geometric intersection region of several (> 2) metrics, the John ellipsoid has to be found by solving an optimization problem [Loseille 2008].

7.5.2 Metric Interpolation

In practice, the metric field is only known discretely at mesh vertices. The definition of an interpolation procedure on metrics is therefore mandatory to be able to compute the metric at any point of the domain. For instance, the computation of the volume of an element using quadrature formula with Relation (7.2) requires the computation of some interpolated metrics inside the considered element.

Several interpolation schemes have been proposed in [Alauzet 2003a] which are based on the simultaneous reduction. The main drawback of these approaches is that the interpolation operation is not commutative. Hence, the result depends on the order in which the operations are performed when more than two metrics are involved. Moreover, such interpolation schemes do not satisfy useful properties such as the maximum principle. Consequently, to design an interpolation scheme on these objects, one needs a consistent operational framework. We suggest to consider the log-Euclidean framework introduced in [Arsigny 2006].

Log-Euclidean framework. We first define the notion of metric logarithm and matrix exponential. The **metric logarithm** is defined on the set of metric tensors. For metric tensor $\mathcal{M} = \mathcal{R}\Lambda^t\mathcal{R}$, it is given by:

$$\ln(\mathcal{M}) := \mathcal{R}\ln(\Lambda)^t\mathcal{R}\,,$$

where $\ln(\Lambda) = diag(\ln(\lambda_i))$. The **matrix exponential** is defined on the set of

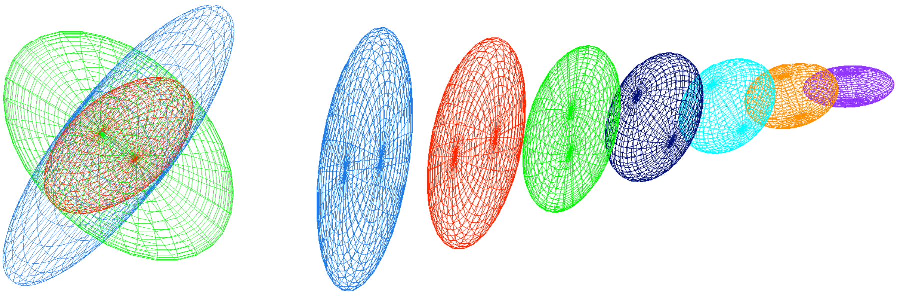


Figure 7.4: *Left, view illustrating the metric intersection procedure with the simultaneous reduction in three dimensions. In red, the resulting metric of the intersection of the blue and green metrics. Right, metric interpolation along a segment where the endpoints metrics are the blue and violet ones.*

symmetric matrices. For any symmetric matrix $\mathcal{S} = \mathcal{Q}\Xi^t\mathcal{Q}$, it is given by:

$$\exp(\mathcal{S}) := \mathcal{Q}\exp(\Xi)^t\mathcal{Q}\,,$$

where $\exp(\Xi) = diag(\exp(\xi_i))$. We can now define the **logarithmic addition** $\oplus$ and the **logarithmic scalar multiplication** $\odot$:

$$\begin{aligned} \mathcal{M}_1 \oplus \mathcal{M}_2 &:= \exp\left(\ln(\mathcal{M}_1) + \ln(\mathcal{M}_2)\right) \\ \alpha \odot \mathcal{M} &:= \exp\left(\alpha.\ln(\mathcal{M})\right) = \mathcal{M}^\alpha\,. \end{aligned}$$

The logarithmic addition is commutative and coincides with matrix multiplication whenever the two tensors $\mathcal{M}_1$ and $\mathcal{M}_2$ commute in the matrix sense. The space of metric tensors, supplied with the logarithmic addition $\oplus$ and the logarithmic scalar multiplication $\odot$ is a vector space.

Remark 2
This framework allows more general computations to be carried out on metric tensors, such as statistical studying or the resolution of PDE's on metric tensors.

Metric interpolation in the log-Euclidean framework. We propose to use the linear interpolation operator derived from the log-Euclidean framework. Let $(\mathbf{x}_i)_{i=1\ldots k}$ be a set of vertices and $(\mathcal{M}(\mathbf{x}_i))_{i=1\ldots k}$ their associated metrics. Then, for a point $\mathbf{x}$ of the domain such that:

$$\mathbf{x} = \sum_{i=1}^{k} \alpha_i.\mathbf{x}_i \quad \text{with} \quad \sum_{i=1}^{k} \alpha_i = 1\,,$$

the interpolated metric is defined by:

$$\mathcal{M}(\mathbf{x}) = \bigoplus_{i=1}^{k} \alpha_i \odot \mathcal{M}(\mathbf{x}_i) = \exp\left(\sum_{i=1}^{k} \alpha_i\, \ln(\mathcal{M}(\mathbf{x}_i))\right)\,. \tag{7.7}$$

This interpolation is commutative, but its bottleneck is to perform k diagonalizations and to request the use of the logarithm and the exponential functions which are CPU consuming. However, this procedure is essential to define continuously the metric map on the entire domain. Moreover, it has been demonstrated in [Arsigny 2006] that this interpolation preserves the maximum principle, *i.e.,* for an edge $\mathbf{pq}$ with endpoints metrics $\mathcal{M}(\mathbf{p})$ and $\mathcal{M}(\mathbf{q})$ such that $\det(\mathcal{M}(\mathbf{p})) < \det(\mathcal{M}(\mathbf{q}))$ then we have $\det(\mathcal{M}(\mathbf{p})) < \det(\mathcal{M}(\mathbf{p} + t\,\mathbf{pq})) < \det(\mathcal{M}(\mathbf{q}))$ for all $t \in [0, 1]$.

Accordingly, this metric interpolation enables a continuous metric field to be defined throughout the entire discretized domain. When a metric is required at a point, we determine to which element the point belongs. Then, we apply Relation (7.7), where α_i are the barycentric coordinates of the point with respect to the element. Figure 7.4 illustrates metric interpolation along a segment, for which the initial data are the endpoints metrics.

Remark 3
The interpolation formulation (7.7) reduces to

$$\mathcal{M}(\mathbf{x}) = \prod_{i=1}^{k} \mathcal{M}(\mathbf{x}_i)^{\alpha_i}\,,$$

if all the metrics commute. Therefore, an arithmetic mean in the log-Euclidean framework could be interpreted as a geometric mean in the space of metric tensors.

7.6 Quantifying mesh anisotropy

In three dimensions, mesh anisotropy can be quantified by two notions:
- the **anisotropic ratios**, and
- the **anisotropic quotients**.

We first recall both notions and the way they are evaluated numerically. Deriving these quantities for an element relies on the fact that there always exists a unique metric tensor for which this element is unit. If $\mathcal{M}_K$ denotes the metric tensor associated with element K, solving the following linear system provides $\mathcal{M}_K$:

$$(S)\ \left\{\begin{array}{l} \ell^2_{\mathcal{M}_K}(\mathbf{e}_1) = 1 \\ \vdots \\ \ell^2_{\mathcal{M}_K}(\mathbf{e}_6) = 1\,, \end{array}\right.$$

where $(\mathbf{e}_i)_{i=1,6}$ is the edges list of K and $\ell^2_{\mathcal{M}_K}(\mathbf{e}_i) = {}^t\mathbf{e}_i\,\mathcal{M}_K\,\mathbf{e}_i$. (S) admits a unique solution as soon as the volume of K is not null. Once $\mathcal{M}_K$ is computed, the anisotropic ratio and the anisotropic quotient associated with element K are simply given by

$$\text{ratio} = \sqrt{\frac{\min_i \lambda_i}{\max_i \lambda_i}} = \frac{\max_i h_i}{\min_i h_i}, \quad \text{and} \quad \text{quo} = \frac{\max_i h_i^3}{h_1 h_2 h_3},$$

where $(\lambda_i)_{i=1,3}$ are the eigenvalues of $\mathcal{M}_K$ and $(h_i)_{i=1,3}$ are the corresponding sizes. The anisotropic ratio stands for the maximum elongation of a tetrahedron by comparing two eigendirections. The anisotropic quotient represents the overall anisotropic ratio of a tetrahedron taking into account all the possible directions. It corresponds to the overall gain in three dimensions of *an anisotropic adapted mesh* as compared to *an isotropic adapted one.* The gain is of course even greater when compared to a uniform mesh.

7.7 Orientation

This chapter has recalled the main techniques related to continuous meshes. Once the metric is known, a class of unit mesh can be defined, and (at least) one unit mesh can be built by the so called *controlled mesh generator*.
Clearly a prerequisite is the definition of the metric from the error estimate. The rest of the dissertation is devoted to this quest.

Chapter 8

Goal-oriented anisotropic mesh adaptation method for unsteady Euler flows

Contents

This chapter has been submitted as a communication co-authored with Frédéric Alauzet and Alain Dervieux to *Journal of Computational Physics.*

8.1 Summary

We present a new algorithm for combining an anisotropic goal-oriented error estimate with the mesh adaptation fixed point method for unsteady problems. The minimization of the error on a functional provides both the density and the anisotropy (stretching) of the optimal mesh. They are expressed in terms of state and adjoint. This method is used for specifying the mesh for a time sub-interval. A global fixed point iterates the re-evaluation of meshes and states over the whole time interval until convergence of the space-time mesh. Applications to unsteady blast-wave and acoustic-wave Euler flows are presented.

8.2 Introduction

Engineering problems frequently require computational fluid dynamics (CFD) solutions with functional outputs of specified accuracy. The computational resources available for these solutions are often limited and errors in solutions and outputs are difficult to control. CFD solutions may be computed with an unnecessarily large number of mesh vertices (and associated high cost) to ensure that the outputs are computed within a required accuracy.

One of the powerful methods for increasing the accuracy and reducing the computational cost is mesh adaptation, the purpose of which is to control the accuracy of the numerical solution by changing the discretization of the computational domain according to mesh size and mesh directions constraints. This technique allows (i) to automatically capture the anisotropy of the physical phenomena, (ii) to substantially reduce the number of degrees of freedom, thus impacting favorably the CPU time, and (iii) to access to high order asymptotic convergence.

The objective of this chapter is to propose a time-accurate anisotropic mesh adaptation method for functional outputs.

Pioneering works ([Peraire 1987]) have shown a fertile development of Hessian-based or metric-based methods [Belhamadia 2004, Bottasso 2004, Dompierre 1997, Frey 2005, Gruau 2005, Hecht 1997, Li 2005, Pain 2001, Tam 2000] which rely on an ideal representation of the *interpolation error* and of the *mesh.* The "multi-scale" version relies on the optimization of the $\mathbf{L^p}$ norm of the interpolation error [Loseille 2009]. It allows to take into account the discontinuities with higher-order convergence [Loseille 2007]. However, these methods are limited to the minimization of some interpolation errors for some solution fields, the "sensors", and do not take into account the PDE being solved. If for many applications, this simplifying standpoint is an advantage, there are also many applications where Hessian-based mesh adaptation is far from optimal regarding the way the degrees of freedom are distributed in the computational domain. Indeed, Hessian-based methods aim at controlling the interpolation error but this purpose is not often so close to the objective that consists in obtaining the best solution of the PDE. Further, in many engineering applications, a specific scalar output needs to be accurately evaluated, e.g. lift, drag, heat flux, and Hessian-based adaptation does not address this issue.

On the other side, *goal-oriented* mesh adaptation focuses on deriving the best mesh to observe a given output functional. Goal-oriented methods result from a series of papers dealing with *a posteriori estimates* (see e.g. [Becker 1996b, Giles 2002b, Verfürth 1996, Wintzer 2008]). But, extracting informations concerning mesh anisotropy from an *a posteriori* estimate is a difficult task. Starting from *a priori estimates*, Loseille *et al.* proposed in [Loseille 2010a] a fully anisotropic goal-oriented mesh adaptation technique for steady problems. This latter method combines goal-oriented rationale and the application of Hessian-based analysis to truncation error.

Mesh adaptation for unsteady flows is also an active field of research and brings an attracting increase in simulation efficiency. Complexity of the algorithms is larger than for steady case: for most flows, the mesh should change during the time interval. Meshes can be moved as in [Baines 1994], pattern-split [Berger 1989, Löhner 1989], locally refined [Alauzet 2011], or globally rebuild as in [Alauzet 2007, Guégan 2010]. Hessian-based methods are essentially applied with a non-moving mesh system.

In this thesis, we do *not* account for *time discretization error* but concentrate on spatial error in unsteady simulations. A mesh adaptation fixed-point method was proposed in [Alauzet 2007]. The Hessian criteria at the different time steps of a sub-interval are synthesized into a single criterion for these steps with the metric intersection [Alauzet 2007, Guégan 2010]. A mesh-PDE solver iteration is applied on time sub-intervals. Extension to $\mathbf{L}^p$ error estimator [Loseille 2007] requires: (i) space-time $\mathbf{L}^\infty - \mathbf{L}^p$ error analysis, (ii) a global fixed-point algorithm to converge the mesh adaptation. This extension has been proposed in [Alauzet 2011].

In the present work, we aim at combining the fully anisotropic goal-oriented mesh adaptation method of [Loseille 2010a] and the fixed-point advances of [Alauzet 2011] for time-accurate mesh adaptation.

To this end, several methodological issues need to be addressed. First, similarly to [Alauzet 2011], we propose a global fixed-point algorithm for solving the coupled system made, this time of three fields, the unsteady state, the unsteady adjoint state and the adapted meshes. Second, this algorithm needs to be *a priori* analyzed and its convergence rate to continuous solution needs to be optimized. Third, at the computer algorithmic level, it is also necessary to master the computational (memory and time) cost of the new system, which couples a time-forward state, a time-backward adjoint and a mesh update influenced by global statistics.

We start this chapter with a formal description of the error analysis in its most general expression, then the application to unsteady compressible Euler flows is presented. In Section 8.5, we introduce the optimal adjoint-based metric definition and all its relative issues, then Section 8.6 is dedicated to strategies for mesh convergence and algorithm optimization. In Section 8.7, we present our mesh adaptation algorithm. This chapter ends with numerical experiments for blast wave problems and acoustic waves.

8.3 Formal Error Analysis

Let us introduce a system of PDE's in its variational formulation:

$$\text{Find } w \in \mathcal{V} \text{ such that } \forall \varphi \in \mathcal{V}, \quad (\Psi(w)\,,\,\varphi) = 0 \tag{8.1}$$

with $\mathcal{V}$ a functional space of solutions.
The associated discrete variational formulation writes then:

$$\text{Find } w_h \in \mathcal{V}_h \text{ such that } \forall \varphi_h \in \mathcal{V}_h, \quad (\Psi_h(w_h)\,,\,\varphi_h) = 0 \tag{8.2}$$

where $\mathcal{V}_h$ is a subspace of $\mathcal{V}$. For a solution w of state system (8.1), we define a *functional output* as:

$$j \in \mathbb{R}\ ;\ j = (g, w), \tag{8.3}$$

where (g, w) holds for the following rather general functional output formulation:

$$(g, w) \ = \ \int_0^T \int_\Omega (g_\Omega, w)\,\mathrm{d}\Omega\,\mathrm{d}t \ + \ \int_\Omega (g_T, w(T))\,\mathrm{d}\Omega \ + \ \int_0^T \int_\Gamma (g_\Gamma, w)\,\mathrm{d}\Gamma\,\mathrm{d}t, \tag{8.4}$$

where g_Ω, g_T, and g_Γ are assumed to be regular enough functions.
We introduce the *continuous adjoint* w^*, solution of the following system:

$$w^* \in \mathcal{V}\ ,\ \forall \psi \in \mathcal{V}\ ,\ \left(\frac{\partial \Psi}{\partial w}(w)\psi, w^* \right) = (g, \psi)\ . \tag{8.5}$$

The objective here is to estimate the following approximation error committed on the functional:

$$\delta j = j(w) - j(w_h)\ ,$$

where w and w_h are respectively solutions of (8.1) and (8.2).

Using the fact that $\mathcal{V}_h \subset \mathcal{V}$, the following error estimates for the unknown can be written:

$$(\Psi_h(w), \varphi_h) - (\Psi_h(w_h), \varphi_h) = (\Psi_h(w), \varphi_h) - (\Psi(w), \varphi_h) = ((\Psi_h - \Psi)(w), \varphi_h). \tag{8.6}$$

It is then useful to choose the test function φ_h as the discrete adjoint state, $\varphi_h = w_h^*$, which is the solution of:

$$\forall \psi_h \in \mathcal{V}_h, \; \left(\frac{\partial \Psi_h}{\partial w_h}(w_h)\psi_h \, , \, w_h^* \right) = (g, \psi_h). \tag{8.7}$$

We assume that w_h^* is close to the continuous adjoint state w^*.

We refer to [Loseille 2010a] in which the following *a priori* formal estimate is finally proposed:

$$\delta j \approx ((\Psi_h - \Psi)(w), w^*) \; . \tag{8.8}$$

The next section is devoted to the application of Estimator (8.8) to the unsteady Euler model.

8.4 Unsteady Euler Models

8.4.1 Continuous state system and Finite Volume formulation

Continuous state system The 3D unsteady compressible Euler equations are set in the computational space-time domain $\mathcal{Q} = \Omega \times [0, T]$, where T is the (positive) maximal time and $\Omega \subset \mathbb{R}^3$ is the spatial domain. An essential ingredient of our discretization and of our analysis is the elementwise linear interpolation operator. In order to use it easily, we define our working functional space as $V = \left[H^1(\Omega) \cap \mathcal{C}(\bar{\Omega}) \right]^5$, that is the set of measurable functions that are continuous with square integrable gradient.

We recall the Euler model in a compact variational formulation in the functional space $\mathcal{V} = H^1\{[0, T]; V\}$:

$$\begin{aligned}
&\text{Find } W \in \mathcal{V} \text{ such that } \forall \varphi \in \mathcal{V}, \quad (\Psi(W) \, , \, \varphi) = 0 \\
\text{with} \quad (\Psi(W) \, , \, \varphi) \;&=\; \int_\Omega \varphi(0)(W_0 - W(0)) \, \mathrm{d}\Omega \;+\; \int_0^T \int_\Omega \varphi W_t \, \mathrm{d}\Omega \, \mathrm{d}t \\
&+\; \int_0^T \int_\Omega \varphi \, \nabla \cdot \mathcal{F}(W) \, \mathrm{d}\Omega \, \mathrm{d}t \;-\; \int_0^T \int_\Gamma \varphi \, \hat{\mathcal{F}}(W).\mathbf{n} \, \mathrm{d}\Gamma \, \mathrm{d}t \, .
\end{aligned} \tag{8.9}$$

Functions φ and W have 5 components, and therefore the product φW holds for $\sum_{k=1..5} \varphi_k W_k$. We have denoted by Γ the boundary of the computational domain Ω, $\mathbf{n}$ is the outward normal to Γ, $W(0)(\mathbf{x}) = W(\mathbf{x}, t)|_{t=0}$ for any $\mathbf{x}$ in Ω, W_0 the initial condition and the boundary flux $\hat{\mathcal{F}}$ contains the different boundary conditions, which involve inflow, outflow and slip boundary conditions.

Discrete state system As a spatially semi-discrete model, we consider the Mixed-Element-Volume formulation of Chapter 2. As in [Loseille 2010a], we reformulate it under the form of a finite element variational formulation, this time in the unsteady context. We assume that Ω is covered by a finite-element partition in simplicial elements denoted K. The mesh, denoted by $\mathcal{H}$ is the set of the elements. Let us introduce the following approximation space:

$$V_h \;=\; \left\{\varphi_h \in V \;\middle|\; \varphi_{h|K} \text{ is affine } \forall K \in \mathcal{H}\right\}, \text{ and } \mathcal{V}_h \;=\; H^1\{[0,T];V_h\} \subset \mathcal{V}.$$

Let Π_h be the usual $\mathcal{P}^1$ projector:

$$\Pi_h : \; V \to \; V_h \;\text{ such that }\; \Pi_h\varphi(\mathbf{x}_i) = \varphi(\mathbf{x}_i), \;\forall\; \mathbf{x}_i \text{ vertex of } \mathcal{H}.$$

We extend it to time-dependent functions:

$$\Pi_h : \; H^1\{[0,T];V\} \to \; \mathcal{V}_h \;\text{ such that }\; \left(\Pi_h\varphi\right)(t) = \Pi_h\left(\varphi(t)\right), \;\forall\; t \in \; [0,T].$$

The weak discrete formulation writes:

$$\text{Find } W_h \in \mathcal{V}_h \text{ such that } \forall\varphi_h \in \mathcal{V}_h, \quad \left(\Psi_h(W_h)\,,\,\varphi_h\right) = 0,$$

$$\begin{aligned}
\text{with: } \left(\Psi_h(W_h)\,,\,\varphi_h\right) \;=\; & \int_\Omega \varphi_h(0)(\Pi_h W_h(0) - W_{0h})\,\mathrm{d}\Omega \;+\; \int_0^T\!\!\int_\Omega \varphi_h\;\Pi_h W_{h,t}\,\mathrm{d}\Omega\,\mathrm{d}t \\
+ & \int_0^T\!\!\int_\Omega \varphi_h \nabla\cdot\mathcal{F}_h(W_h)\,\mathrm{d}\Omega\,\mathrm{d}t \;-\; \int_0^T\!\!\int_\Gamma \varphi_h\hat{\mathcal{F}}_h(W_h).\mathbf{n}\,\mathrm{d}\Gamma\,\mathrm{d}t \\
+ & \int_0^T\!\!\int_\Omega \varphi_h\, D_h(W_h)\,\mathrm{d}\Omega\,\mathrm{d}t,
\end{aligned} \tag{8.10}$$

with $\mathcal{F}_h = \Pi_h\mathcal{F}$ and $\hat{\mathcal{F}}_h = \Pi_h\hat{\mathcal{F}}$. The D_h term accounts for the numerical diffusion. In short, it involves the difference between the Galerkin central-differences approximation and a second-order Godunov approximation [Cournède 2006]. In the present study, we only need to know that for smooth fields, the D_h term is a third order term with respect to the mesh size. For shocked fields, monotonicity limiters become first-order terms.

Practical experiments are done with the in-house CFD software `Wolf`. The numerical scheme is vertex-centered and uses a particular edge-based formulation. This formulation consists in associating with each vertex of the mesh a control volume (or Finite-Volume cell) built by the rule of medians. This flow solver uses a HLLC approximate Riemann solver to compute numerical fluxes. A high-order scheme is derived according to a MUSCL type method using downstream and upstream tetrahedra. Appropriate β-schemes are adopted for the variable extrapolation which gives us a very high-order space-accurate scheme for the linear advection on cartesian triangular meshes. This approach provides low diffusion second-order space-accurate scheme in the non-linear case. The MUSCL type method is combined with a generalization of the Superbee limiter with three entries to guarantee the TVD property of the scheme. An explicit time stepping algorithm is used by means of multi-stages, high-order strong-stability-preserving (SSP) Runge-Kutta scheme. More details can be found in [Alauzet 2010a].

8.4.2 Continuous Adjoint system and discretization

Continuous adjoint system We refer here to the continuous adjoint System (8.5) introduced previously:

$$W^* \in \mathcal{V}\ ,\ \forall \psi \in \mathcal{V}\ :\ \left(\frac{\partial \Psi}{\partial W}(W)\psi, W^*\right) - (g,\psi)\ =\ 0. \tag{8.11}$$

We recall that (g,ψ) is defined by (8.4).

Replacing $\Psi(W)$ by its Formulation (8.9) and integrating by parts, we get:

$$\begin{aligned}\left(\frac{\partial \Psi}{\partial W}(W)\psi, W^*\right) \ &=\ \int_\Omega (\psi(0)W^*(0) - \psi(T)W^*(T))\,\mathrm{d}\Omega \\ &+\ \int_0^T\int_\Omega \psi\left(-W_t^* - \left(\frac{\partial \mathcal{F}}{\partial W}\right)^*\nabla W^*\right)\mathrm{d}\Omega\,\mathrm{d}t \\ &+\ \int_0^T\int_\Gamma \psi\left[\left(\frac{\partial \mathcal{F}}{\partial W}\right)^* W^*.\mathbf{n}\ -\ \left(\frac{\partial \hat{\mathcal{F}}}{\partial W}\right)^* W^*.\mathbf{n}\right]\mathrm{d}\Gamma\,\mathrm{d}t\ .\end{aligned} \tag{8.12}$$

Consequently, the continuous adjoint state W^* must be such that:

$$-\,W_t^* - \left(\frac{\partial \mathcal{F}}{\partial W}\right)^*\nabla W^*\ =\ g_\Omega\ \text{ in }\ \Omega \tag{8.13}$$

with the associated adjoint boundary conditions:

$$\left(\frac{\partial \mathcal{F}}{\partial W}\right)^* W^*.\mathbf{n}\ -\ \left(\frac{\partial \hat{\mathcal{F}}}{\partial W}\right)^* W^*.\mathbf{n}\ =\ g_\Gamma\ \text{ on }\ \Gamma$$

and the final adjoint state condition:

$$W^*(T)\ =\ g_T.$$

The adjoint Euler equations is a system of advection equations, where the temporal integration goes backwards, *i.e.,* in the opposite direction of usual time. Thus, when solving the unsteady adjoint system, one starts at the end of the flow run and progresses back until reaching the start time.

Discrete adjoint system Although any consistent approximation of the continuous adjoint system could be built by discretizing System (8.13), we choose the option to build the discrete adjoint system from the discrete state system defined by Relation (8.10) in order to be closer to the true error from which the continuous model is derived.

Consider the following semi-discrete unsteady compressible Euler model (explicit RK1 time integration):

$$\Psi_h^n(W_h^n, W_h^{n-1}) = \frac{W_h^n - W_h^{n-1}}{\delta t^n} + \Phi_h(W_h^{n-1}) = 0 \quad \text{for } n = 1, ..., N. \tag{8.14}$$

The time-dependent functional is discretized as follows:

$$j_h(W_h) = \sum_{n=1}^{N} \delta t^n j_h^{n-1}(W_h^{n-1}).$$

For the sake of simplicity, we restrict to the case $g_T = 0$ for the functional output defined by Relation (8.4).

The problem of minimizing the error committed on the target functional $j(W_h) = (g, W_h)$, subject to the Euler system (8.14), can be transformed into an unconstrained problem for the following Lagrangian functional [Giles 2000]:

$$\mathcal{L}(W_h, W_h^*) = \sum_{n=1}^{N} \delta t^n j_h^{n-1}(W_h^{n-1}) - \sum_{n=1}^{N} \delta t^n (W_h^{*,n})^T \Psi_h^n(W_h^n, W_h^{n-1}),$$

where $W_h^{*,n}$ are the N vectors of the Lagrange multipliers (which are the time-dependent adjoint states).
The conditions for an extremum are:

$$\frac{\partial \mathcal{L}}{\partial W_h^{*,n}} = 0 \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial W_h^{n}} = 0, \quad \text{for } n = 1, ..., N.$$

The first condition is clearly verified from Relation (8.14).
Thus the Lagrangian multipliers $W_h^{*,n}$ must be chosen such that the second condition of extrema is verified. This provides the unsteady discrete adjoint system:

$$\left\{ \begin{array}{lcl} W_h^{*,N} & = & 0 \\ W_h^{*,n-1} & = & W_h^{*,n} + \delta t^n \dfrac{\partial j_h^{n-1}}{\partial W_h^{n-1}}(W_h^{n-1}) - \delta t^n (W_h^{*,n})^T \dfrac{\partial \Phi_h}{\partial W_h^{n-1}}(W_h^{n-1}), \end{array} \right. \tag{8.15}$$

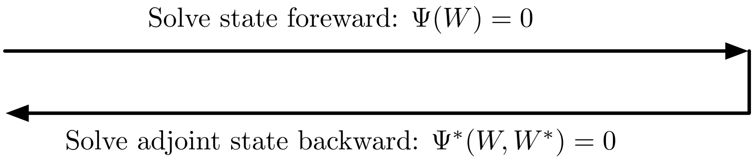
or equivalently, the semi-discrete unsteady adjoint model reads:

$$\Psi_h^{*,n}(W_h^{*,n}, W_h^{*,n-1}, W_h^{n-1}) = \frac{W_h^{*,n-1} - W_h^{*,n}}{-\delta t^n} + \Phi_h^*(W_h^{*,n}, W_h^{n-1}) = 0 \quad \text{for } n = 1, ..., N$$

with

$$\Phi_h^*(W_h^{*,n}, W_h^{n-1}) = \frac{\partial j_h^{n-1}}{\partial W_h^{n-1}}(W_h^{n-1}) - (W_h^{*,n})^T \frac{\partial \Phi_h}{\partial W_h^{n-1}}(W_h^{n-1}).$$

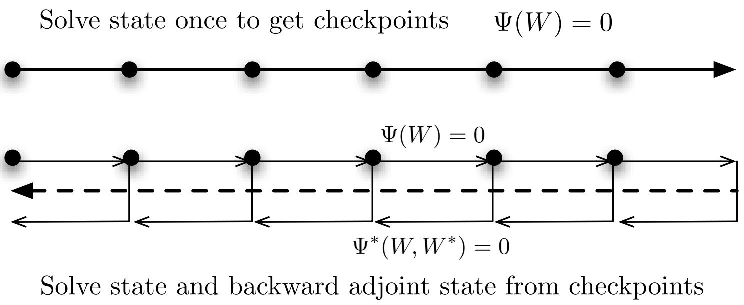
As the adjoint system runs in reverse time, the first expression in the adjoint System (8.15) is referred to as adjoint "initialization".


Solve state foreward: $\Psi(W) = 0$
Solve adjoint state backward: $\Psi^*(W, W^*) = 0$

Computing $W_h^{*,n-1}$ at time t^{n-1} requires the knowledge of state W_h^{n-1} and adjoint state $W_h^{*,n}$. Therefore, the knowledge of all states $\left\{W_h^{n-1}\right\}_{n=1,N}$ is needed to compute backward the adjoint state from time T to 0 which involves large memory storage effort. For instance, if we consider a 3D simulation with a mesh composed of one million vertices then we need to store at each iteration five millions solution data (we have 5 conservative variables). If we perform 1000 iterations, then the memory effort to store all states is 37.25 Gb for double-type data storage (or 18.62 for float-type data storage). Two strategies are employed to reduce importantly this drawback: checkpoints and interpolation.

The memory effort can be reduced by out-of-core storage of checkpoints as shown in the picture below. First the state-simulation is performed to store checkpoints. Second, when computing backward the adjoint, we first recompute all states from the checkpoint and store them in memory and then we compute the unsteady adjoint until the checkpoint physical time. This method implies a recomputing effort of the state W.

The other strategy consists in storing solution states in memory only each m solver iterations. When the unsteady adjoint is solved, solution states between two savings are linearly interpolated. This method leads to a loss of accuracy for the unsteady adjoint computation.


Solve state once to get checkpoints
$\Psi(W) = 0$
$\Psi(W) = 0$
$\Psi^*(W, W^*) = 0$
Solve state and backward adjoint state from checkpoints

8.4.3 Impact of the adjoint: numerical example

Before going more deeply in the error model, we would like to emphasize how strongly the use of an adjoint may impact the density distribution of adapted meshes.

The simulation of a blast in a 2D geometry representing a city is performed, see Figure 8.1. A blast-like initialization $W_{blast} = (10, 0, 0, 250)$ in ambient air $W_{air} = (1, 0, 0, 2.5)$ is considered in a small region of the computational domain. We perform a forward/backward computation on a uniform mesh of 22 574 vertices and 44 415 triangles.
Output functional of interest j is the quadratic deviation from ambient pressure on target surface S which is a part of the higher building roof (Figure 8.1):

$$j(W) = \int_0^T \int_S \frac{1}{2}(p(t) - p_{air})^2 \,\mathrm{d}S \,\mathrm{d}t.$$

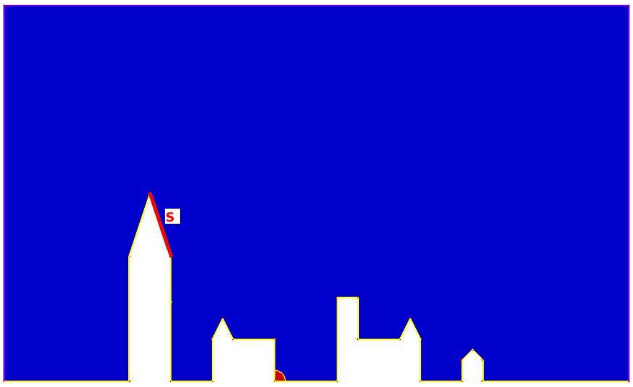


Figure 8.1: Initial blast solution and location of target surface S.

Figure 8.2 plots the density isolines of the flow at different times showing several shock waves traveling throughout the computational domain. Figure 8.3 depicts the associated density adjoint state progressing backward in time. The same a-dimensional physical time is considered for both figures.

The simulation points out the ability of the adjoint to automatically provide the sensitivity of the flow field on the functional. Indeed, at early time of the simulation (top left picture), a lot of information is captured by the adjoint, *i.e.,* non-zero adjoint values. We notice that shock waves which will directly impact the targeted surface are clearly detected by the adjoint, but also shocks waves reflected by the left building which will be redirected towards surface S.
At the middle of the simulation, the adjoint neglects waves that are traveling in the direction opposite to S and also waves that will not impact surface S before final time T since they won't have an influence on the cost functional. While getting closer to final time T (bottom right picture), the adjoint only focuses on the last waves that will impact surface S and ignores the rest of the flow.

8.5 Optimal unsteady adjoint-based metric

8.5.1 Error analysis (applied to unsteady Euler model)

We replace in Estimation (8.8) operators Ψ and Ψ_h by their expressions given by Relations (8.9) and (8.10). In [Loseille 2010a], it was observed that even for shocked flows, it is interesting to neglect the numerical viscosity term. We follow again this option. We also discard the error committed when imposing the initial condition.

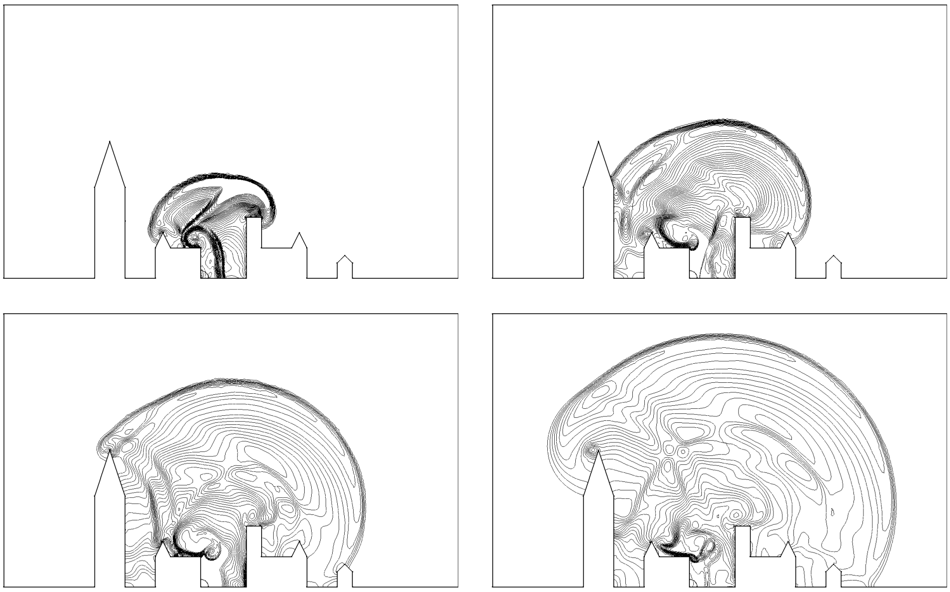


Figure 8.2: 2D city blast solution state evolution. From left to right and top to bottom, snapshot of the density isolines at a-dimensional time 1.2, 2.25, 3.3 and 4.35.

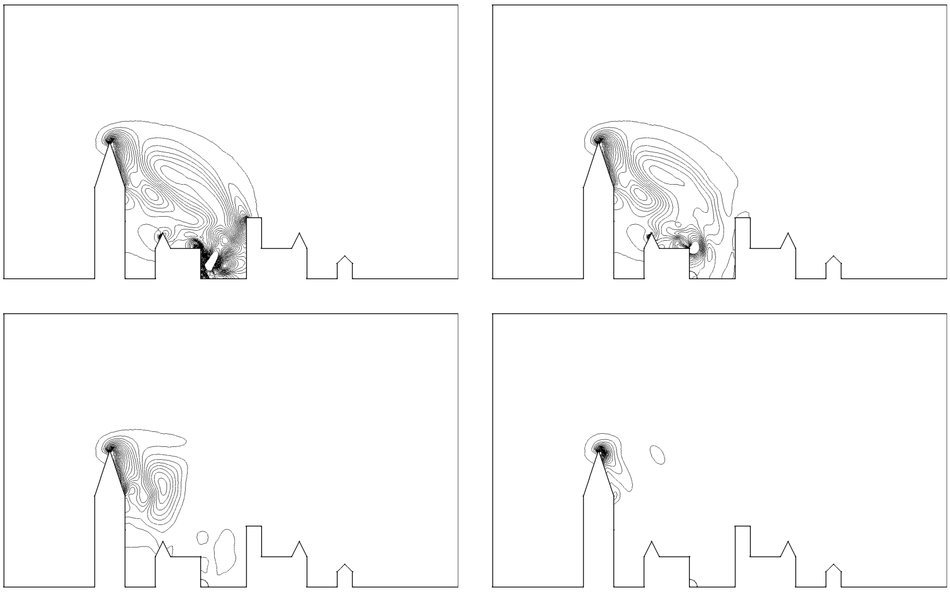


Figure 8.3: 2D city blast adjoint state evolution. From left to right and top to bottom, snapshot of the adjoint-density isolines at a-dimensional time 1.2, 2.25, 3.3 and 4.35.

We finally get the following simplified error model:

$$\begin{aligned}\delta j \;\approx\;& \int_0^T\int_\Omega W^*\left(W-\Pi_h W\right)_t \mathrm{d}\Omega\,\mathrm{d}t \;+\; \int_0^T\int_\Omega W^*\,\nabla.\big(\mathcal{F}(W)-\Pi_h\mathcal{F}(W)\big)\,\mathrm{d}\Omega\,\mathrm{d}t\\ -\;& \int_0^T\int_\Gamma W^*\left(\hat{\mathcal{F}}(W)-\Pi_h\hat{\mathcal{F}}(W))\right).\mathbf{n}\,\mathrm{d}\Gamma\,\mathrm{d}t\,. \qquad (8.16)\end{aligned}$$

Integrating by parts leads to:

$$
\begin{aligned}
\delta j \quad \approx \quad & \int_0^T \int_\Omega W^* \left(W - \Pi_h W\right)_t \mathrm{d}\Omega\,\mathrm{d}t \; - \; \int_0^T \int_\Omega \nabla W^* \left(\mathcal{F}(W) - \Pi_h \mathcal{F}(W)\right) \mathrm{d}\Omega\,\mathrm{d}t \\
- \; & \int_0^T \int_\Gamma W^* \left(\bar{\mathcal{F}}(W) - \Pi_h \bar{\mathcal{F}}(W))\right).\mathbf{n}\,\mathrm{d}\Gamma\,\mathrm{d}t\,. \qquad (8.17)
\end{aligned}
$$

with $\bar{\mathcal{F}} = \hat{\mathcal{F}} - \mathcal{F}$.

We observe that this estimate of δj is expressed in terms of interpolation errors of the Euler fluxes and of the time derivative weighted by continuous functions W^* and ∇W^*.

Error bound with a safety principle The integrands in Error Estimation (9.4) contain positive and negative parts which can compensate for some particular meshes. In our strategy, we prefer to not rely on these parasitic effects and to slightly over-estimate the error. To this end, all integrands are bounded by their absolute values:

$$
\begin{aligned}
(g, W_h - W) \quad \leq \quad & \int_0^T \int_\Omega |W^*| \left|\left(W - \Pi_h W\right)_t\right| \mathrm{d}\Omega\,\mathrm{d}t \\
+ \quad & \int_0^T \int_\Omega |\nabla W^*| \left|\mathcal{F}(W) - \Pi_h \mathcal{F}(W)\right| \mathrm{d}\Omega\,\mathrm{d}t \\
+ \quad & \int_0^T \int_\Gamma |W^*| \left|(\bar{\mathcal{F}}(W) - \Pi_h \bar{\mathcal{F}}(W)).\mathbf{n}\right| \mathrm{d}\Gamma\,\mathrm{d}t\,. \qquad (8.18)
\end{aligned}
$$

8.5.2 Continuous mesh model

We propose to work in the continuous mesh framework, introduced in [Loseille 2011a, Loseille 2011b] and described shortly in Chapter 7.

The main idea of this framework is to model continuously discrete meshes by Riemannian metric spaces. It allows us to define proper differentiable optimization [Absil 2008, Arsigny 2006], *i.e.,* to use a calculus of variations on continuous meshes which cannot apply on the class of discrete meshes. This framework lies in the class of metric-based methods.
We recall that a continuous mesh $\mathbf{M}$ of computational domain Ω is identified to a Riemannian metric field [Berger 2003] $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. For all $\mathbf{x}$ of Ω, $\mathcal{M}(\mathbf{x})$ is a symmetric 3×3 matrix having $(\lambda_i(\mathbf{x}))_{i=1,3}$ as eigenvalues along the principal directions $\mathcal{R}(\mathbf{x}) = (\mathbf{v}_i(\mathbf{x}))_{i=1,3}$. Sizes along these directions are denoted $(h_i(\mathbf{x}))_{i=1,3} = (\lambda_i^{-\frac{1}{2}}(\mathbf{x}))_{i=1,3}$ and the three *anisotropy quotients* r_i are defined by: $r_i = h_i^3\,(h_1 h_2 h_3)^{-1}$.

The diagonalisation of $\mathcal{M}(\mathbf{x})$ writes:

$$\mathcal{M}(\mathbf{x}) = d^{\frac{2}{3}}(\mathbf{x})\,\mathcal{R}(\mathbf{x}) \begin{pmatrix} r_1^{-\frac{2}{3}}(\mathbf{x}) & & \\ & r_2^{-\frac{2}{3}}(\mathbf{x}) & \\ & & r_3^{-\frac{2}{3}}(\mathbf{x}) \end{pmatrix} {}^t\mathcal{R}(\mathbf{x}), \tag{8.19}$$

The *node density* d is equal to: $d = (h_1 h_2 h_3)^{-1} = (\lambda_1 \lambda_2 \lambda_3)^{\frac{1}{2}} = \sqrt{\det(\mathcal{M})}$.
By integrating the node density, we define the *complexity* $\mathcal{C}$ of a continuous mesh which is the continuous counterpart of the total number of vertices:

$$\mathcal{C}(\mathbf{M}) = \int_\Omega d(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}.$$

Given a continuous mesh $\mathbf{M}$, we shall say, following [Loseille 2011a, Loseille 2011b], that a discrete mesh $\mathcal{H}$ of the same domain Ω is a **unit mesh with respect to M**, if each tetrahedron $K \in \mathcal{H}$, defined by its list of edges $(\mathbf{e}_i)_{i=1\ldots6}$, verifies:

$$\forall i \in [1,6], \quad \ell_\mathcal{M}(\mathbf{e}_i) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right] \quad \text{and} \quad Q_\mathcal{M}(K) \in [\alpha, 1] \ \text{ with } \ \alpha > 0\,,$$

in which the length of an edge $\ell_\mathcal{M}(\mathbf{e}_i)$ and the quality of an element $Q_\mathcal{M}(K)$ are defined as follows:

$$Q_\mathcal{M}(K) = \frac{36}{3^{\frac{1}{3}}} \frac{|K|_\mathcal{M}^{\frac{2}{3}}}{\sum_{i=1}^6 \ell^2_\mathcal{M}(\mathbf{e}_i)} \in [0,1], \ \text{with } |K|_\mathcal{M} = \int_K \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x},$$
$$\text{and } \ \ell_\mathcal{M}(\mathbf{e}_i) = \int_0^1 \sqrt{{}^t\mathbf{ab}\ \mathcal{M}(\mathbf{a} + t\,\mathbf{ab})\ \mathbf{ab}}\,\mathrm{d}t, \quad \text{with } \mathbf{e}_i = \mathbf{ab}.$$

We choose a tolerance α equal to 0.8.

We want to emphasize that the set of all the discrete meshes that are unit meshes with respect to a unique $\mathbf{M}$ contains an infinite number of meshes. Given a smooth function u, to each unit mesh $\mathcal{H}$ with respect to $\mathbf{M}$ corresponds a local interpolation error $|u - \Pi u|$. In [Loseille 2011a, Loseille 2011b], it is shown that all these interpolation errors are well represented by the so-called continuous interpolation error related to $\mathbf{M}$, which is expressed locally in terms of the Hessian H_u of u as follows:

$$\begin{aligned}(u - \pi_\mathcal{M} u)(\mathbf{x},t) &= \frac{1}{10}\mathrm{trace}(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\,|H_u(\mathbf{x},t)|\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x}))\\ &= \frac{1}{10} d(\mathbf{x})^{-\frac{2}{3}} \sum_{i=1}^{3} r_i(\mathbf{x})^{\frac{2}{3}}\, {}^t\mathbf{v_i}(\mathbf{x})\,|H_u(\mathbf{x},t)|\,\mathbf{v_i}(\mathbf{x}),\end{aligned} \tag{8.20}$$

where $|H_u|$ is deduced from H_u by taking the absolute values of its eigenvalues and where time-dependency notations have been added for use in next sections.

8.5.3 Continuous error model

Working in this framework enables us to write Estimate (8.18) in a continuous form:

$$\begin{aligned} |(g, W_h - W)| \approx \mathbf{E}(\mathbf{M}) &= \int_0^T \int_\Omega |W^*| \, \left|\left(W - \pi_{\mathcal{M}} W\right)_t\right| \mathrm{d}\Omega \, \mathrm{d}t \\ &+ \int_0^T \int_\Omega |\nabla W^*| \, |\mathcal{F}(W) - \pi_{\mathcal{M}} \mathcal{F}(W)| \, \mathrm{d}\Omega \, \mathrm{d}t \\ &+ \int_0^T \int_\Gamma |W^*| \, |(\bar{\mathcal{F}}(W) - \pi_{\mathcal{M}} \bar{\mathcal{F}}(W)).\mathbf{n}| \, \mathrm{d}\Gamma \, \mathrm{d}t. \end{aligned} \tag{8.21}$$

We observe that the third term introduce a dependency of the error with respect to the boundary surface mesh. In the present paper, we discard this term and refer to [Loseille 2010a] for a discussion of the influence of it. Then, introducing the continuous interpolation error, we can write the simplified error model as follows:

$$\mathbf{E}(\mathbf{M}) = \int_0^T \int_\Omega \operatorname{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\, \mathbf{H}(\mathbf{x},t)\, \mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\right) \mathrm{d}\Omega \, \mathrm{d}t$$

$$\text{with} \quad \mathbf{H}(\mathbf{x},t) = \sum_{j=1}^{5} \left([\Delta t]_j(\mathbf{x},t) + [\Delta x]_j(\mathbf{x},t) + [\Delta y]_j(\mathbf{x},t) + [\Delta z]_j(\mathbf{x},t)\right), \tag{8.22}$$

in which :

$$\begin{aligned} [\Delta t]_j(\mathbf{x},t) &= \left|W_j^*(\mathbf{x},t)\right| \cdot \left|H(W_{j,t})(\mathbf{x},t)\right|, \\ [\Delta x]_j(\mathbf{x},t) &= \left|\frac{\partial W_j^*}{\partial x}(\mathbf{x},t)\right| \cdot \left|H(\mathcal{F}_1(W_j))(\mathbf{x},t)\right|, \\ [\Delta y]_j(\mathbf{x},t) &= \left|\frac{\partial W_j^*}{\partial y}(\mathbf{x},t)\right| \cdot \left|H(\mathcal{F}_2(W_j))(\mathbf{x},t)\right|, \\ [\Delta z]_j(\mathbf{x},t) &= \left|\frac{\partial W_j^*}{\partial z}(\mathbf{x},t)\right| \cdot \left|H(\mathcal{F}_3(W_j))(\mathbf{x},t)\right|. \end{aligned}$$

Here, W_j^* denotes the j^{th} component of the adjoint vector W^*, $H(\mathcal{F}_i(W_j))$ the Hessian of the j^{th} component of the vector $\mathcal{F}_i(W)$, and $H(W_{j,t})$ the Hessian of the j^{th} component of the time derivative of W.

The mesh optimization problem writes:

$$\text{Find } \mathbf{M}_{opt} = \text{Argmin}_{\mathbf{M}} \, \mathbf{E}(\mathbf{M}), \tag{8.23}$$

under the constraint of bounded mesh fineness:

$$\mathcal{C}_{st}(\mathbf{M}) = N_{st}, \tag{8.24}$$

where N_{st} is a specified total number of nodes.
Since we consider an unsteady problem, the space-time (st) complexity used to compute the solution takes into account the time discretization. The above constraint then imposes the total number of nodes in the time integral, that is:

$$\mathcal{C}_{st}(\mathbf{M}) = \int_0^T \tau(t)^{-1} \left(\int_\Omega d_{\mathcal{M}}(\mathbf{x},t) \mathrm{d}\mathbf{x} \right) \mathrm{d}t \tag{8.25}$$

where $\tau(t)$ is the time step used at time t of interval $[0,T]$.

8.5.4 Spatial minimization for a fixed t

Let us assume that at time t, we seek for the optimal continuous mesh $\mathbf{M}_{go}(t)$ which minimizes the instantaneous error, *i.e.,* the spatial error for a fixed time t:

$$\tilde{\mathbf{E}}(\mathbf{M}(t)) = \int_\Omega \mathrm{trace} \left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\, \mathbf{H}(\mathbf{x},t)\, \mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t) \right) \mathrm{d}\mathbf{x}$$

under the constraint that the number of vertices is prescribed to

$$\mathcal{C}(\mathbf{M}(t)) = \int_\Omega d_{\mathcal{M}(t)}(\mathbf{x},t)\, \mathrm{d}\mathbf{x} = N(t). \tag{8.26}$$

Similarly to [Loseille 2010a], solving the optimality conditions provides the *optimal goal-oriented ("go") instantaneous continuous mesh* $\mathbf{M}_{go}(t) = (\mathcal{M}_{go}(\mathbf{x},t))_{\mathbf{x}\in\Omega}$ at time t defined by:

$$\mathcal{M}_{go}(\mathbf{x},t) = N(t)^{\frac{2}{3}}\, \mathcal{M}_{go,1}(\mathbf{x},t)\,, \tag{8.27}$$

where $\mathcal{M}_{go,1}$ is the optimum for $\mathcal{C}(\mathbf{M}(t)) = 1$:

$$\mathcal{M}_{go,1}(\mathbf{x},t) = \left(\int_\Omega (\det \mathbf{H}(\bar{\mathbf{x}},t))^{\frac{1}{5}} \mathrm{d}\bar{\mathbf{x}} \right)^{-\frac{2}{3}} (\det \mathbf{H}(\mathbf{x},t))^{-\frac{1}{5}}\; \mathbf{H}(\mathbf{x},t). \tag{8.28}$$

The corresponding optimal instantaneous error at time t writes:

$$\tilde{\mathbf{E}}(\mathbf{M}_{go}(t)) = 3\, N(t)^{-\frac{2}{3}} \left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{1}{5}} \mathrm{d}\mathbf{x} \right)^{\frac{5}{3}} = 3\, N(t)^{-\frac{2}{3}}\, \mathcal{K}(t)\,. \tag{8.29}$$

For the sequel, we denote: $\mathcal{K}(t) = \left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{1}{5}} \mathrm{d}\mathbf{x} \right)^{\frac{5}{3}}$.

8.5.5 Temporal minimization

To complete the resolution of optimization Problem (8.23-8.24), we perform a temporal minimization in order to get the optimal space-time continuous mesh. In other words, we need to find the optimal time law $t \to N(t)$ for the instantaneous mesh size.
First, we consider the simpler case where the time step τ is specified by the user as a function of time $t \to \tau(t)$. Second, we deal with the case of an explicit time advancing solver subject to Courant time step condition.

Temporal minimization for specified τ Let us consider the case where the time step τ is specified by a function of time $t \to \tau(t)$. After the spatial optimization, the space-time error writes:

$$\mathbf{E}(\mathbf{M}_{go}) = \int_0^T \tilde{\mathbf{E}}(\mathbf{M}_{go}(t))\,\mathrm{d}t = 3\int_0^T N(t)^{-\frac{2}{3}}\,\mathcal{K}(t)\,\mathrm{d}t \tag{8.30}$$

and we aim at minimizing it under the following space-time complexity constraint:

$$\int_0^T N(t)\tau(t)^{-1}\,\mathrm{d}t = N_{st}. \tag{8.31}$$

In other words, we concentrate on seeking for *the optimal distribution of $N(t)$ when the space-time total number of nodes N_{st} is prescribed.*
Let us apply the one-to-one change of variables:

$$\tilde{N}(t) = N(t)\tau(t)^{-1} \quad \text{and} \quad \tilde{\mathcal{K}}(t) = \tau(t)^{-\frac{2}{3}}\,\mathcal{K}(t)\,.$$

Then, our temporal optimization problem becomes:

$$\min_{\mathbf{M}}\ \mathbf{E}(\mathbf{M}) = \int_0^T \tilde{N}(t)^{-\frac{2}{3}}\,\tilde{\mathcal{K}}(t)\,\mathrm{d}t \quad \text{under constraint} \quad \int_0^T \tilde{N}(t)\,\mathrm{d}t = N_{st}\,.$$

The solution of this problem is given by:

$$\tilde{N}_{opt}(t)^{-\frac{5}{3}}\,\tilde{\mathcal{K}}(t) = const \quad \Rightarrow \quad N_{opt}(t) = C(N_{st})\ (\tau(t)\,\mathcal{K}(t))^{\frac{3}{5}}\,.$$

Here, constant $C(N_{st})$ can be obtained by introducing the above expression in space-time complexity Constraint (8.31), leading to:

$$C(N_{st}) = \left(\int_0^T \tau(t)^{-\frac{2}{5}}\,\mathcal{K}(t)^{\frac{3}{5}}\mathrm{d}t\right)^{-1} N_{st}\,,$$

which completes the description of the optimal space-time metric for a prescribed time step.

Using Relation (8.27), the analytic expression of the optimal space-time goal-oriented metric $\mathbf{M}_{go}$ writes:

$$\mathcal{M}_{go}(\mathbf{x},t) = N_{st}^{\frac{2}{3}}\left(\int_0^T \tau(t)^{-\frac{2}{5}}\left(\int_\Omega (\det\mathbf{H}(\bar{\mathbf{x}},t))^{\frac{1}{5}}\mathrm{d}\bar{\mathbf{x}}\right)\mathrm{d}t\right)^{-\frac{2}{3}} \tau(t)^{\frac{2}{5}}(\det\mathbf{H}(\mathbf{x},t))^{-\frac{1}{5}}\mathbf{H}(\mathbf{x},t)\,. \tag{8.32}$$

We get the following optimal error:

$$\mathbf{E}(\mathbf{M}_{go}) = 3\,N_{st}^{-\frac{2}{3}}\ \left(\int_0^T \tau(t)^{-\frac{2}{5}}\left(\int_\Omega (\det\mathbf{H}(\mathbf{x},t))^{\frac{1}{5}}\mathrm{d}\mathbf{x}\right)\mathrm{d}t\right)^{\frac{5}{3}}\,. \tag{8.33}$$

Temporal minimization for explicit time advancing In the case of an explicit time advancing subject to a Courant condition, we get a more complex context, since time step strongly depends on the smallest mesh size. We restrict to the case of smooth data and solution.

We still seek for the optimal continuous mesh that minimizes space-time Error (8.30) under complexity Constraint (8.31).

Let $\Delta x_{min,1}(t) = \min_{\mathbf{x}} \min_i h_i(\mathbf{x})$ be the smallest mesh size of $\mathbf{M}_{go,1}(t)$. Since the metric Definition (8.19) is homogeneous with the inverse square of mesh size, we deduce the smallest mesh size of $\mathbf{M}_{go}(t)$ from Relation (8.27):

$$\Delta x_{min}(t) = N(t)^{-\frac{1}{3}}\, \Delta x_{min,1}(t),$$

where $\Delta x_{min,1}(t)$ is independent of the mesh complexity.
A way to write the Courant condition for time-advancing is to define the time step $\tau(t)$ by:

$$\tau(t) = c(t)^{-1}\, \Delta x_{min}(t) = N(t)^{-\frac{1}{3}}\, c(t)^{-1}\, \Delta x_{min,1}(t)\,,$$

where $c(t)$ is the maximal wave speed over the domain at time t.
Again, we search for the optimal distribution of $N(t)$ when the space-time complexity N_{st} is prescribed (Relation (8.31)), with

$$N_{st} = \int_0^T N(t)^{\frac{4}{3}}\, c(t)\, (\Delta x_{min,1}(t))^{-1}\, \mathrm{d}t\,.$$

We choose to apply the one-to-one change of variables:

$$\hat{N}(t) = N(t)^{\frac{4}{3}}\, c(t)\, (\Delta x_{min,1}(t))^{-1} \quad \text{and} \quad \hat{\mathcal{K}}(t) = \mathcal{K}(t)\, c(t)^{\frac{1}{2}}\, (\Delta x_{min,1}(t))^{-\frac{1}{2}}\,.$$

Therefore, the corresponding space-time approximation error over the simulation time interval and space-time complexity reduces to:

$$\mathbf{E}(\mathbf{M}_{go}) = 3\int_0^T N(t)^{-\frac{2}{3}}\, \mathcal{K}(t)\, \mathrm{d}t = 3\int_0^T \hat{N}(t)^{-\frac{1}{2}} \hat{\mathcal{K}}(t)\, \mathrm{d}t \quad \text{and} \quad \int_0^T \hat{N}(t) \mathrm{d}t = N_{st}\,.$$

This optimization problem has for solution:

$$\hat{N}_{opt}(t)^{-\frac{3}{2}} \hat{\mathcal{K}}(t) = const \quad \Rightarrow \quad \hat{N}_{opt}(t) = C(N_{st})\ \hat{\mathcal{K}}(t)^{\frac{2}{3}}$$

and by considering the space-time complexity constraint relation we deduce:

$$C(N_{st}) = N_{st}\ \left(\int_0^T \hat{\mathcal{K}}(t)^{\frac{2}{3}}\, \mathrm{d}t \right)^{-1}\,.$$

Using the definitions of $\hat{N}$ and $\hat{\mathcal{K}}$ in the above relations, we get:

$$\begin{aligned} N(t)^{\frac{4}{3}}\, c(t)\, (\Delta x_{min,1}(t))^{-1} \quad &= \quad N_{st}\ \left(\int_0^T \left(\mathcal{K}(t)\, c(t)^{\frac{1}{2}}\, (\Delta x_{min,1}(t))^{-\frac{1}{2}} \right)^{\frac{2}{3}} \mathrm{d}t \right)^{-1} \\ &\cdot \quad \left(\mathcal{K}(t)\, c(t)^{\frac{1}{2}}\, (\Delta x_{min,1}(t))^{-\frac{1}{2}} \right)^{\frac{2}{3}} \end{aligned}$$

$$\Longleftrightarrow N(t) = N_{st}^{\frac{3}{4}}\, c(t)^{-\frac{1}{2}}\, (\Delta x_{min,1}(t))^{\frac{1}{2}} \mathcal{K}(t)^{\frac{1}{2}} \left(\int_0^T c(t)^{\frac{1}{3}}\, (\Delta x_{min,1}(t))^{-\frac{1}{3}} \mathcal{K}(t)^{\frac{2}{3}}\, \mathrm{d}t \right)^{-\frac{3}{4}} .$$

Consequently, after some simplifications, we obtain the following expression of the optimal space-time goal-oriented continuous mesh $\mathbf{M}_{go}$ and error:

$$\mathcal{M}_{go}(\mathbf{x},t) = N_{st}^{\frac{1}{2}} \left(\int_0^T \theta(t)^{\frac{1}{3}}\, \mathcal{K}(t)^{\frac{2}{3}}\, \mathrm{d}t \right)^{-\frac{1}{2}} \theta(t)^{-\frac{1}{3}}\, \mathcal{K}(t)^{-\frac{1}{15}} (\det \mathbf{H}(\mathbf{x},t))^{-\frac{1}{5}} \mathbf{H}(\mathbf{x},t) \tag{8.34}$$

$$\mathbf{E}(\mathbf{M}_{go}) = 3\, N_{st}^{-\frac{1}{2}} \left(\int_0^T \theta(t)^{\frac{1}{3}}\, \mathcal{K}(t)^{\frac{2}{3}}\, \mathrm{d}t \right)^{\frac{3}{2}} , \tag{8.35}$$

where $\theta(t) = c(t)\, (\Delta x_{min,1}(t))^{-1}$ and $\mathcal{K}(t) = \left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{1}{5}} \mathrm{d}\mathbf{x} \right)^{\frac{5}{3}}$.

8.5.6 Temporal minimization for time sub-intervals

The previous analysis provides the optimal size of the adapted meshes for each time level. Hence, this analysis requires the mesh to be adapted at each flow solver time step. But, in practice this approach involves a very large number of remeshing which is CPU consuming and spoils solution accuracy due to many solution transfers from one mesh to a new one. In consequence, a new adaptive strategy has been proposed in [Alauzet 2007, Alauzet 2011] where the number of remeshing is controlled (thus drastically reduced) by generating adapted meshes for several solver time steps.

The idea is to split the simulation time interval into n_{adap} *sub-intervals* $[t_{i-1}, t_i]$ for $i = 1, .., n_{adap}$. Each spatial mesh $\mathbf{M}^i$ is then kept constant during each sub-interval $[t_{i-1}, t_i]$.

We could consider this partition as a *time discretization of the mesh adaptation problem*. In other words, the number of nodes N^i of the i^{th} adapted mesh $\mathbf{M}^i$ on sub-interval $[t_{i-1}, t_i]$ should for example be taken equal to:

$$N^i = \frac{\int_{t_{i-1}}^{t_i} N_{opt}(t) \tau(t)^{-1} \mathrm{d}t}{\int_{t_{i-1}}^{t_i} \tau(t)^{-1} \mathrm{d}t}.$$

Here, we propose a different option in which we get an optimal discrete answer.

Spatial minimization on a sub-interval Given the continuous mesh complexity N^i for the single adapted mesh used during time sub-interval $[t_{i-1}, t_i]$, we seek for the optimal continuous mesh $\mathbf{M}^i_{go}$ solution of the following problem:

$$\min_{\mathbf{M}^i} \mathbf{E}^i(\mathbf{M}^i) = \int_\Omega \operatorname{trace}\left((\mathcal{M}^i)^{-\frac{1}{2}}(\mathbf{x})\, \mathbf{H}^i(\mathbf{x})\, (\mathcal{M}^i)^{-\frac{1}{2}}(\mathbf{x}) \right) \mathrm{d}\mathbf{x} \quad \text{such that} \quad \mathcal{C}(\mathbf{M}^i) = N^i ,$$

where matrix $\mathbf{H}^i$ on the sub-interval can be defined by either using an $\mathbf{L}^1$ or an $\mathbf{L}^\infty$ norm:

$$\mathbf{H}^i_{\mathbf{L}^1}(\mathbf{x}) = \int_{t_{i-1}}^{t_i} \mathbf{H}(\mathbf{x},t)\,\mathrm{d}t \quad \text{or} \quad \mathbf{H}^i_{\mathbf{L}^\infty}(\mathbf{x}) = \Delta t_i \max_{t\in[t_{i-1},t_i]} \mathbf{H}(\mathbf{x},t)\,,$$

with $\Delta t_i = t_i - t_{i-1}$.

Processing as previously, we get the spatial optimality condition:

$$\mathcal{M}^i_{go}(\mathbf{x}) = (N^i)^{\frac{2}{3}}\,\mathcal{M}^i_{go,1}(\mathbf{x}) \quad \text{with } \mathcal{M}^i_{go,1}(\mathbf{x}) = \left(\int_\Omega (\det \mathbf{H}^i(\bar{\mathbf{x}}))^{\frac{1}{5}}\mathrm{d}\bar{\mathbf{x}}\right)^{-\frac{2}{3}} (\det \mathbf{H}^i(\mathbf{x}))^{-\frac{1}{5}}\,\mathbf{H}^i(\mathbf{x}).$$

The corresponding optimal error $\mathbf{E}^i(\mathbf{M}^i_{go})$ writes:

$$\mathbf{E}^i(\mathbf{M}^i_{go}) = 3\,(N^i)^{-\frac{2}{3}} \left(\int_\Omega (\det \mathbf{H}^i(\mathbf{x}))^{\frac{1}{5}}\mathrm{d}\mathbf{x}\right)^{\frac{5}{3}} = 3\,(N^i)^{-\frac{2}{3}}\,\mathcal{K}^i\,.$$

To complete our analysis, we shall perform a temporal minimization. Again, we first consider the case where the time step τ is specified by a function of time and, secondly, we deal then with the case of an explicit time advancing solver subject to Courant time step condition.

Temporal minimization for specified τ After the spatial minimization, the temporal optimization problem reads:

$$\min_{\mathbf{M}} \mathbf{E}(\mathbf{M}) = \sum_{i=1}^{n_{adap}} \mathbf{E}^i(\mathbf{M}^i_{go}) = 3 \sum_{i=1}^{n_{adap}} (N^i)^{-\frac{2}{3}}\,\mathcal{K}^i$$

such that

$$\sum_{i=1}^{n_{adap}} N^i \left(\int_{t_{i-1}}^{t_i} \tau(t)^{-1}\mathrm{d}t\right) = N_{st}\,.$$

We set the one-to-one mapping:

$$\tilde{N}^i = N^i \left(\int_{t_{i-1}}^{t_i} \tau(t)^{-1}\mathrm{d}t\right) \quad \text{and} \quad \tilde{\mathcal{K}}^i = \mathcal{K}^i \left(\int_{t_{i-1}}^{t_i} \tau(t)^{-1}\mathrm{d}t\right)^{\frac{2}{3}}\,,$$

then the optimization problem reduces to:

$$\min_{\mathbf{M}} \sum_{i=1}^{n_{adap}} (\tilde{N}^i)^{-\frac{2}{3}}\,\tilde{\mathcal{K}}^i \quad \text{such that} \quad \sum_{i=1}^{n_{adap}} \tilde{N}^i = N_{st}\,.$$

The solution is:

$$\tilde{N}^i_{opt} = \mathcal{C}(N_{st})\,(\tilde{\mathcal{K}}^i)^{\frac{3}{5}} \quad \text{with} \quad \mathcal{C}(N_{st}) = N_{st}\left(\sum_{i=1}^{n_{adap}} (\tilde{\mathcal{K}}^i)^{\frac{3}{5}}\right)^{-1}$$

$$\Rightarrow \quad N^i = N_{st}\left(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{3}{5}} \left(\int_{t_{i-1}}^{t_i} \tau(t)^{-1}\mathrm{d}t\right)^{\frac{2}{5}}\right)^{-1} (\mathcal{K}^i)^{\frac{3}{5}} \left(\int_{t_{i-1}}^{t_i} \tau(t)^{-1}\mathrm{d}t\right)^{-\frac{3}{5}}\,.$$

and we deduce the following optimal continuous mesh $\mathbf{M}_{go} = \{\mathbf{M}_{go}^i\}_{i=1,..,n_{adap}}$ and error:

$$\mathcal{M}_{go}^i(\mathbf{x}) \,=\, N_{st}^{\frac{2}{3}}\Bigg(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{3}{5}}\Big(\int_{t_{i-1}}^{t_i} \tau(t)^{-1}\mathrm{d}t\Big)^{\frac{2}{5}}\Bigg)^{-\frac{2}{3}}\Big(\int_{t_{i-1}}^{t_i} \tau(t)^{-1}\mathrm{d}t\Big)^{-\frac{2}{5}}(\det \mathbf{H}^i(\mathbf{x}))^{-\frac{1}{5}}\mathbf{H}^i(\mathbf{x}) \tag{8.36}$$

$$\mathbf{E}(\mathbf{M}_{go}) \,=\, 3\, N_{st}^{-\frac{2}{3}}\Bigg(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{3}{5}}\,\Big(\int_{t_{i-1}}^{t_i} \tau(t)^{-1}\mathrm{d}t\Big)^{\frac{2}{5}}\Bigg)^{\frac{5}{3}}, \tag{8.37}$$

with $(\mathcal{K}^i)^{\frac{3}{5}} = \displaystyle\int_\Omega (\det \mathbf{H}^i(\mathbf{x}))^{\frac{1}{5}}\mathrm{d}\mathbf{x}$.

Temporal minimization for explicit time advancing Similarly to the previous section, the Courant-based time-step writes:

$$\tau(t) = c(t)^{-1}\,\Delta x_{min}^i = (N^i)^{-\frac{1}{3}}\,c(t)^{-1}\,\Delta x_{min,1}^i \quad \text{for } t \in [t_{i-1}, t_i]\,,$$

where $\Delta x_{min,1}^i$ is the smallest altitude of $\mathbf{M}_{go,1}^i$ and $c(t)$ is the maximal wave speed over the domain.
The optimization problem writes:

$$\min_{\mathbf{M}} \mathbf{E}(\mathbf{M}) = \sum_{i=1}^{n_{adap}} \mathbf{E}^i(\mathbf{M}_{go}^i) = 3\,\sum_{i=1}^{n_{adap}} (N^i)^{-\frac{2}{3}}\,\mathcal{K}^i$$

under the constraint:

$$\sum_{i=1}^{n_{adap}} (N^i)^{\frac{4}{3}}\,\left(\int_{t_{i-1}}^{t_i} c(t)\,(\Delta x_{min,1}^i)^{-1}\mathrm{d}t\right) = N_{st}\,.$$

We set again:

$$\hat{N}^i = \left(N^i\right)^{\frac{4}{3}}\left(\int_{t_{i-1}}^{t_i} c(t)\,(\Delta x_{min,1}^i)^{-1}\,\mathrm{d}t\right) \quad\text{and}\quad \hat{\mathcal{K}}^i = \mathcal{K}^i\left(\int_{t_{i-1}}^{t_i} c(t)\,(\Delta x_{min,1}^i)^{-1}\,\mathrm{d}t\right)^{\frac{1}{2}}.$$

Then, the optimization problem becomes:

$$\min_{\mathbf{M}} \mathbf{E}(\mathbf{M}) = \sum_{i=1}^{n_{adap}} \mathbf{E}^i(\mathbf{M}_{go}^i) = 3\,\sum_{i=1}^{n_{adap}} (N^i)^{-\frac{2}{3}}\,\mathcal{K}^i = 3\,\sum_{i=1}^{n_{adap}} (\hat{N}^i)^{-\frac{1}{2}}\,\hat{\mathcal{K}}^i$$

under the constraint:

$$\sum_{i=1}^{n_{adap}} (N^i)^{\frac{4}{3}}\,\left(\int_{t_{i-1}}^{t_i} c(t)\,(\Delta x_{min,1}^i)^{-1}\mathrm{d}t\right) = \sum_{i=1}^{n_{adap}} \hat{N}^i = N_{st}\,.$$

This optimization problem has for solution:

$$\hat{N}_{opt}^i = \mathcal{C}(N_{st})\,(\hat{\mathcal{K}}^i)^{\frac{2}{3}} \quad\text{with}\quad \mathcal{C}(N_{st}) = N_{st}\left(\sum_{i=1}^{n_{adap}} (\hat{\mathcal{K}}^i)^{\frac{2}{3}}\right)^{-1},$$

from which we deduce:

$$N^i_{opt} = N_{st}^{\frac{3}{4}} \left(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{2}{3}} \left(\int_{t_{i-1}}^{t_i} c(t)\, (\Delta x^i_{min,1})^{-1} \mathrm{d}t \right)^{\frac{1}{3}} \right)^{-\frac{3}{4}} (\mathcal{K}^i)^{\frac{1}{2}} \left(\int_{t_{i-1}}^{t_i} c(t)\, (\Delta x^i_{min,1})^{-1} \mathrm{d}t \right)^{-\frac{1}{2}}.$$

For the sake of clarity, we set: $\theta^i = \int_{t_{i-1}}^{t_i} c(t)\,(\Delta x^i_{min,1})^{-1}\mathrm{d}t$.

The optimal continuous mesh $\mathbf{M}_{go} = \{\mathbf{M}^i_{go}\}_{i=1,..,n_{adap}}$ and error read:

$$\mathcal{M}^i_{go}(\mathbf{x}) \;=\; N_{st}^{\frac{1}{2}} \left(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{2}{3}} (\theta^i)^{\frac{1}{3}} \right)^{-\frac{1}{2}} (\theta^i)^{-\frac{1}{3}} (\mathcal{K}^i)^{-\frac{1}{15}} (\det \mathbf{H}^i(\mathbf{x}))^{-\frac{1}{5}} \mathbf{H}^i(\mathbf{x}) \tag{8.38}$$

$$\mathbf{E}(\mathbf{M}_{go}) \;=\; 3\, N_{st}^{-\frac{1}{2}} \left(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{2}{3}} (\theta^i)^{\frac{1}{3}} \right)^{\frac{3}{2}}. \tag{8.39}$$

Remark 1: The choice of the time sub-intervals $\{[t_{i-1}, t_i]\}_{i=1,n_{adap}}$ for a given n_{adap} is a mesh adaptation problem: what is the subdivision of interval $[0,T]$ in n_{adap} sub-intervals which will minimize the error on optimal mesh/metric $\mathcal{M}$? Since we take a constant metric in the sub-interval, the error main term in approximating $\mathcal{M}$ is the following integral of the absolute value of the time-derivative of $\mathcal{M}$:

$$\sum_{i=1}^{n_{adap}} \int_{t_{i-1}}^{t_i} \left| \frac{\partial \mathcal{M}(t)}{\partial t} \right| \mathrm{d}t.$$

which can be minimized for instance by isodistribution of the error by sub-interval.

Remark 2: The parameter n_{adap}, number of time sub-intervals is important for the efficiency of the adaptation algorithm. When a too large value is prescribed for n_{adap}, the error in mesh-to-mesh interpolation may increase. A compromise needs to be found by the user.

8.6 Theoretical Mesh Convergence Analysis

Our motivation in developing mesh adaptation algorithms is to get approximation algorithms with better convergence to continuous target data. By better, we mean that we want to reach the asymptotic high order convergence with a lower number of nodes and also for solutions involving discontinuities.

Both improvements have been previously obtained in the context of steady inviscid flows [Alauzet 2010a]. The present work focuses on mesh adaptation only controlling the spatial approximation error in the context of unsteady flows. From this standpoint, we just forget about time approximation error by specifying a time step or by considering an explicit time advancing context close to the one of references [Alauzet 2007, Alauzet 2011, Guégan 2010]. We assume that accuracy of the time advancing scheme is second-order at least. Then it can be derived from basic arguments that the time approximation error is essentially smaller than or equal to the spatial approximation error, controlled by the metric-based method, which justifies to adapt the mesh only to spatial error [Alauzet 2007, Alauzet 2011, Guégan 2010].

In order to measure the theoretical convergence order of the mesh adaptation algorithm, we need to compare the global mesh effort -the complexity N_{st}- with the corresponding error level. We recall that an adaptative or an uniform discretisation approach both using N degrees of freedom has a convergence of order α if the approximation error $|u - u_N|$ satisfies:

$$|u - u_N| \leq const.\ N^{-\frac{\alpha}{d}}$$

where d is the computational domain Euclidian dimension.

In the following, the convergence order of numerical solutions computed with the presented adaptive platform is established for all the different strategies described above. First, the case of smooth flows is given, then the case of singular flows is exemplified by the case of a traveling discontinuity.

8.6.1 Smooth flow fields

For some Hessian-based methods, *i.e.,* the $\mathbf{L}^\infty - \mathbf{L}^p$ approach of [Guégan 2010], an analysis is proposed for predicting the order of convergence to the continuous solution. In the goal-oriented method discussed in this chapter, the adaptive state solution W_h generally does *not* converge to the continuous one W. However, in the case of regular solutions, the expression of the optimal error model indicates that the functional output indeed converges, and with a predictable order.

Smooth context with specified time step Here, we are adapting the mesh with the purpose of reducing the spatial error. The space dimension is 3. Now, in this case, we have established the following estimate:

$$\mathbf{E}(\mathbf{M}_{go}) = O(N_{st}^{-\frac{2}{3}}) \quad \text{as} \quad N_{st} \to \infty.$$

for the case of an adaptation at each time step (8.33) and also for the case of an adaptation with a fixed mesh at each sub-interval (8.37), therefore:

Lemma 8.6.1
The modeled spatial error on functional for a specified time-step converges at second-order rate.

Smooth context with Courant-based time step According to the argument recalled above, we are adapting the mesh $\mathbf{M}(t)$ in order to, by the magic of Courant condition, reduce both space and time error. The space-time dimension is 4. Now, in this case, we have established the following estimate:

$$\mathbf{E}(\mathbf{M}_{go}) = O(N_{st}^{-\frac{1}{2}}) \quad \text{as} \quad N_{st} \to \infty.$$

for the case of an adaptation at each time step, (8.35), and also for the case of an adaptation with a fixed mesh at each sub-interval, (8.39), therefore:.

Lemma 8.6.2
The space-time modeled error on functional for Courant-based time step converges at second-order rate.

8.6.2 Singular flow fields

An important advantage of mesh adaptation is its potential ability in addressing with a better mesh convergence the case of solution fields involving (isolated) singularities such as discontinuities, etc.

In this case, it can happen that the specified mesh density become locally infinitely large and the mesh size can be zero. In the unsteady case, the time step becomes also zero or too small. Not only the time advancing is stopped or too slow, but also the evaluation of the global effort (defined by the mesh size divided by the time step) becomes difficult to use. This can be avoided by forcing the mesh size to be larger than a given length. We do not consider this issue in this section.

We propose a short analysis dealing with the case of a *prescribed time step* and suggesting a few rules for attempting to have second-order *spatial* convergence.

According to remarks of [Loseille 2007], second-order space convergence can be obtained *also* for singular case with discontinuities, assuming that the error on singularity is concentrated on a subset of zero measure of the computational domain to be discretized by the mesh. For simplicity, we consider only the error committed when we simply *interpolate* a given function on a mesh.

Steady flow with a shock The $\mathbf{L}^1$ convergence of a piecewise linear interpolation is of second order far from the shock. It is only first-order on the shock. In order to reduce the error by a factor four, mesh-size normal to the shock needs to be reduced by a factor four.

In several papers, as in [Loseille 2007], we have demonstrated that a good anisotropic mesh adaptation algorithm allows us to still get second-order convergence. This is made possible by imposing that the region near singularity, where mesh size should be four times smaller, can itself have its measure reduced by a factor four. Therefore, it is possible to apply the following convergence strategy (3D case):

- mesh size outside shock is divided by 2 in the three space directions and the number of nodes is multiplied by 8. Taking into account the second-order accuracy of the considered interpolation, this produce a local error 4 times smaller,
- the thickness of the region around shock is divided by 4. In the new region around the shock, mesh size is divided by 2 parallel to shock and by 4 normal to shock. Although the density of nodes is multiplied by 16, the number of nodes is multiplied by 4, thanks to the reduction of thickness. Consequently, the overall number of nodes is essentially multiplied by 8 for an error divided by 4.

Lemma 8.6.3
For a steady flow field involving a shock, interpolation error can be divided by a factor 4 *with a total number of nodes multiplied by* 8*, which expresses the second-order spatial convergence. This property does not only hold for anisotropic mesh adaptive interpolation, but also for anisotropic mesh adaptive algorithms for PDEs, see [Loseille 2007].*

Unsteady flow with a moving shock The space-wise second-order capture of the singularity is well addressed by the spatial mesh adaptation: for each time level, we apply a metric-based anisotropic spatial mesh adaptation.
Mesh size normal to the discontinuity will be reduced by a factor 4 only in the vicinity of the discontinuity. In other directions near discontinuity and in any direction in other regions of the computational domain, the mesh size is divided by 2.

In 3D, with this strategy, passing from N spatial degrees of freedom to $8N$ leads to a four times reduction of the spatial error. Unfortunately, the time step also needs to be reduced by a factor 4, leading to a total number of degrees of freedom multiplied by 32. Thus:

$$||u - u_{32N}||_{L^1(\Omega\times[0,T])} \leq \frac{1}{4}||u - u_N||_{L^1(\Omega\times[0,T])} \,. \tag{8.40}$$

By comparing $\frac{1}{4}$ with $32^{-\alpha/n}$, this time with $n = 4$ (space-time dimension), we get the following barrier convergence lemma:

Lemma 8.6.4
Barrier convergence order for time advancing discretizations. *For a time-advancing second-order flow solver coupled with an unsteady goal-oriented mesh adaptation at each time level applied to a traveling discontinuity, the space-time convergence rate is at most $\alpha = 8/5$.*

Remark 3: The limitation at rate 8/5 applies to the cases where both smooth regions and discontinuities are present. In the case of a flow involving only discontinuities and no smooth region, the spatial mesh size can be bounded at $4N_{tot}$ and the second order can be recovered. In the case without discontinuity, the second order is automatically recovered because the time step is only reduced by a factor 2.

In fact, we are not only using a time advancing space-time mesh, but we are also considering the fixed-point mesh adaptation algorithm. The situation is (slightly) worse since we are working with meshes that are prismatic in time, that is, made of the product of a single spatial mesh by several time levels. This single spatial mesh needs to account for the *motion* of any singularity.

If the singularity, for example a shock wave, moves during the simulation, then the region of the spatial mesh where the singularity progresses is the *union* of all singularity neighboring regions for any time level of the time interval. We call this region the trajectory of the discontinuity. In contrast to the steady case, this region is a *thick* region.

At convergence, the measure of this thick region swept by the moving discontinuity is essentially proportional to its normal velocity times the time sub-interval width ΔT. For a single time sub-interval of fixed width, this measure cannot tend to zero, hence a large number of nodes is wasted. When we apply a process of convergence to the continuous, we divide by 4 the mesh size normal to discontinuity and by 2 in the other directions. With an unchanged time sub-interval width, the number of nodes in the trajectory of the discontinuity is 16 times larger and 8 times larger for the rest of the mesh. It is thus mandatory to decrease time sub-interval width ΔT by a factor 2 (*i.e.,* to increase n_{adap} accordingly in the case where $\Delta T = T/n_{adap}$) to recover higher order convergence. In a rather artificial manner, we can analyze the spatial convergence order by considering only the mean number of nodes, quotient of total space-time number of nodes by the number of time steps: $N_{mean} = N_{st}/n_{\mathcal{T}}$.

Lemma 8.6.5
Spatial convergence for the transient fixed-point mesh adaptation. *For an unsteady flow field with a moving shock, a necessary condition for second order convergence in space is that the time sub-interval width is divided by 2 when smooth region mesh size is divided by 2 in each of the three space directions and when singular region mesh size is divided by 2 in directions parallel to shock and by 4 in direction normal to shock. To synthesize, in the case of a uniform division into time sub-intervals of size $\Delta T = T/n_{adap}$, we expect to get a spatial second-order spatial convergence in $\mathbf{L}^1([0,T];\mathbf{L}^1(\Omega))$ by passing from (N_{mean}, n_{adap}) to $(8N_{mean}, 2n_{adap})$.* □

Indeed, in such a case, the density of nodes is locally multiplied by 16 in the thick trajectory of the discontinuity but the thickness of this region is divided by 2 leading to a total number of nodes multiplied by 4, the main mesh effort comes from the smooth region.

The more realistic evaluation of space-time convergence order needs to account for the number or equivalently the size of the time step. According to Lemma 5.4, we cannot hope better than a convergence at order 8/5. We divide the time step by

4, since the traveling discontinuity is also a timewise discontinuity. This increases by a factor 4 the space-time number of nodes which needs also to be increased by a factor 8 at each time level. We also need to divide the time sub-interval length ΔT by 2 for limiting the number of nodes in the vicinity of the discontinuity trajectory.

Lemma 8.6.6
Space-time convergence for the transient fixed-point mesh adaptation. *We consider a second-order flow solver coupled with an unsteady goal-oriented mesh adaptation applied to a 3D traveling discontinuity. We assume a uniform division of the simulation time frame into time sub-intervals of size $\Delta T = T/n_{adap}$. Then, we expect a space-time convergence in $\mathbf{L}^1([0,T];\mathbf{L}^1(\Omega))$ at order 8/5 by dividing the time step by a factor 4: $\tau \to \tau/4$ and passing from (N_{st}, n_{adap}) to $(32N_{st}, 2n_{adap})$.*

Remark 4: Adapting for a time sub-interval instead of adapting at each time steps does not degrade the asymptotic convergence order which is a very good news. Nevertheless, such a series of adapted meshes is only space-time sub-optimal as the constant in the error term is larger than the adaptation at each time step.

Remark 5: Between sub-intervals, transfers of the solution fields from the previous mesh to the new one are necessary. The choice of the transfer operator has certainly some impact on the global accuracy (see for example [Alauzet 2010b]) together with how frequently it is applied. Reference [Alauzet 2010b] suggests to consider conservative interpolation for compressible flows.

8.7 From theory to practice

The continuous mesh adaptation problem takes the form of the following continuous optimality conditions:

$$\begin{aligned} &W \in \mathcal{V}\ ,\ \forall \varphi \in \mathcal{V}\ ,\ \ (\Psi(\mathcal{M}, W), \varphi) = 0 && \text{“Euler system”} \\ &W^* \in \mathcal{V}\ ,\ \forall \psi \in \mathcal{V}\ ,\ \left(\frac{\partial \Psi}{\partial W}(\mathcal{M}, W)\psi, W^*\right) = (g, \psi) && \text{“adjoint system”} \\ &\mathcal{M}(\mathbf{x}, t) = \mathcal{M}_{go}(\mathbf{x}, t) && \text{“adapted mesh”.} \end{aligned}$$

In practice, it remains to approximate the above three-field coupled system by a discrete one and then solve it. For discretising the state and adjoint PDE's, we take the spatial schemes introduced previously which are explicit Runge-Kutta time advancing schemes. Such time discretization methods have non-linear stability properties like TVD which are particularly suitable for the integration of system of hyperbolic conservation laws where discontinuities appear. Discretising the last equation consists in specifying the mesh according to a discrete metric deduced from the discrete states.

In order to remedy all the problematics relative to mesh adaptation for time-dependent simulations stated in the introduction, an innovative strategy based on a fixed-point algorithm has been initiated in [Alauzet 2003b] and fully developed in [Alauzet 2007].

The fixed-point algorithm aims at avoiding the generation of a new mesh at each solver iteration which would imply serious degradation of the CPU time and of the solution accuracy due to the large number of mesh modifications. It is also an answer to the lag problem occurring when, from the solution at t^n, a new adapted mesh is generated at level t^n to compute next solution at level t^{n+1}. In that latter case, since the mesh is not adapted to the solution evolution between time levels t^n and t^{n+1}, the mesh is always late with respect to the physics.

The fixed point approach has been successfully applied to bi-fluids three-dimensional problems [Guégan 2010], to a blast simulation in a three-dimensional city [Alauzet 2007] and to moving bodies simulations [Alauzet 2011].

The basic idea consists in splitting the simulation time frame $[0, T]$ into n_{adap} adaptation sub-intervals:

$$[0, T] \;=\; [0 = t_0, \, t_1] \cup \ldots \cup [t_i, \, t_{i+1}] \cup \ldots \cup [t_{n_{adap}-1}, \, t_{n_{adap}}] \, ,$$

and to keep the same adapted mesh for each sub-interval. Consequently, the time-dependent simulation is performed with only n_{adap} different adapted meshes. The mesh used on each sub-interval is adapted to control the solution accuracy from t_{i-1} to t_i. We examine now how to apply this program.

8.7.1 Choice of the goal-oriented metric

The optimal adapted meshes for each sub-interval are generated according to analysis of Section 8.5.6. In this work, the following particular choice has been made:

- the Hessian metric for sub-interval i is based on a control of the temporal error in $\mathbf{L}^\infty$ norm:
$$\mathbf{H}^i_{\mathbf{L}^\infty}(\mathbf{x}) = \Delta t_i \max_{t \in [t_i, t_{i+1}]} \mathbf{H}(\mathbf{x}, t) = \Delta t_i \, \mathbf{H}^i_{\max}(\mathbf{x}) \, ,$$
- function $\tau : t \to \tau(t)$ is constant and equal to 1,
- all sub-intervals have the same time length Δt.

The optimal goal-oriented metric $\mathbf{M}_{go} = \{\mathbf{M}^i_{go}\}_{i=1,..,n_{adap}}$ then simplifies to:

$$\mathcal{M}^i_{go}(\mathbf{x}) = N_{st}^{\frac{2}{3}} \left(\sum_{i=1}^{n_{adap}} (\int_\Omega (\det \mathbf{H}^i_{\max}(\mathbf{x}))^{\frac{1}{5}} \mathrm{d}\mathbf{x}) \right)^{-\frac{2}{3}} (\Delta t)^{\frac{1}{3}} \left(\det \mathbf{H}^i_{\max}(\mathbf{x})\right)^{-\frac{1}{5}} \mathbf{H}^i_{\max}(\mathbf{x}) \, .$$

Remark 7: We notice that we obtain a similar expression of the optimal metric to the one proposed in [Alauzet 2011], but it is presently obtained in a goal-oriented context and by means of a space-time error minimization.

8.7.2 Global fixed-point mesh adaptation algorithm

To converge the non-linear mesh adaptation problem, *i.e.,* converging the couple mesh-solution, we propose a fixed-point mesh adaptation algorithm. The parameter N_{st} representing the total computational effort is prescribed by the user and will influence the size of all the meshes defined during the time interval. That is, to compute any metric fields $\mathbf{M}^i_{go}$, we have to evaluate a global normalization factor which requires the knowledge of all $\mathbf{H}^i_{\max}$. Thus, the whole simulation from 0 to T must be performed to be able to evaluate all metrics $\mathbf{M}^i_{go}$.

Similarly to [Alauzet 2011], a global fixed point strategy covering the whole time-frame $[0, T]$, called *Global adjoint fixed-point mesh adaptation algorithm*, is considered:

```
//- Fixed-point loop to converge global space-time mesh adaptation
For j=1,$n_{nptfx}$
  //- Solve state once to get checkpoints
  For i=1,$n_{adap}$
    • $\mathcal{W}^j_{0,i}$ = ConservativeSolutionTransfer($\mathcal{H}^j_{i-1}, \mathcal{W}^j_{i-1}, \mathcal{H}^j_i$)
    • $\mathcal{W}^j_i$ = SolveStateForward($\mathcal{W}^j_{0,i}, \mathcal{H}^j_i$)
  End for
  //- Solve state and adjoint backward and store samples
  For i=$n_{adap}$, 1
    • $(\mathcal{W}^*)^j_i$ = AdjointStateTransfer($\mathcal{H}^j_{i+1}, (\mathcal{W}^*_0)^j_{i+1}, \mathcal{H}^j_i$)
    • $\{\mathcal{W}^j_i(k), (\mathcal{W}^*)^j_i(k)\}_{k=1,n_k}$ = SolveStateAndAdjointBackward($\mathcal{W}^j_{0,i}, (\mathcal{W}^*)^j_i, \mathcal{H}^j_i$)
    • $|\mathbf{H}_{\max}|^j_i$ = ComputeGoalOrientedHessianMetric($\mathcal{H}^j_i, \{\mathcal{W}^j_i(k), (\mathcal{W}^*)^j_i(k)\}_{k=1,n_k}$)
  End for
  • $\mathcal{C}^j$ = ComputeSpaceTimeComplexity($\{|\mathbf{H}_{\max}|^j_i\}_{i=1,n_{adap}}$)
  • $\{\mathcal{M}^j_i\}_{i=1,n_{adap}}$ = ComputeUnsteadyGoalOrientedMetrics($\mathcal{C}^j, \{|\mathbf{H}_{\max}|^j_i\}_{i=1,n_{adap}}$)
  • $\{\mathcal{H}^{j+1}_i\}_{i=1,n_{adap}}$ = GenerateAdaptedMeshes($\{\mathcal{H}^j_i\}_{i=1,n_{adap}}, \{\mathcal{M}^j_i\}_{i=1,n_{adap}}$)
End for
```

Let us describe this algorithm sketched in Figure 8.4. It consists in splitting the time interval $[0, T]$ into the n_{adap} mesh-adaptation time sub-intervals: $\{[t_{i-1}, t_i]\}_{i=1,..,n_{adap}}$ with $t_0 = 0$ and $t_{n_{adap}} = T$. On each sub-interval a different mesh is used. A time-forward computation of the state solution is first performed with a out-of-core storage of all checkpoints, which are taken to be $\{\mathcal{W}_h(t_i)\}_{i=1,..,n_{adap}}$. Between each sub-interval, the solution is interpolated on the new mesh using the conservative interpolation of [Alauzet 2010b]. Then, starting from the last sub-interval and proceeding until the first one, we recompute and store in memory all solution states of the sub-interval from the previously stored checkpoint in order to evaluate time-backward the adjoint state throughout the sub-interval. At the same time, we evaluate the Hessian metrics $\mathbf{H}^i_{\max}$ required to generate new adapted meshes for each sub-interval. To this end, n_k Hessian metric sample are computed on each sub-interval and intersected [Alauzet 2007] to obtain $\mathbf{H}^i_{\max}$. At the end of the computation, the global space-time mesh complexity is evaluated, producing weights for the goal-oriented metric fields for each sub-interval. Finally, all new adapted meshes are generated according to the prescribed metrics. The time-forward, time-backward and mesh update steps are repeated into the $j = 1, .., n_{ptfx}$ global fixed-point loop. Convergence of the fixed-point is obtained in typically 5 global iterations.

This mesh adaptation loop has been fully parallelized. The solution transfer, the solver and the Hessians computation have been parallelized using a p-thread paradigm at the element loop level [Alauzet 2009]. As regards the computation of the metrics and the generation of the adapted meshes, we observe that they can be achieved in a decoupled manner once the complete simulation has been performed, leading to an easy parallelization of these stages. Indeed, if n_{adap} processors are available, each metric and mesh can be done on one processor.

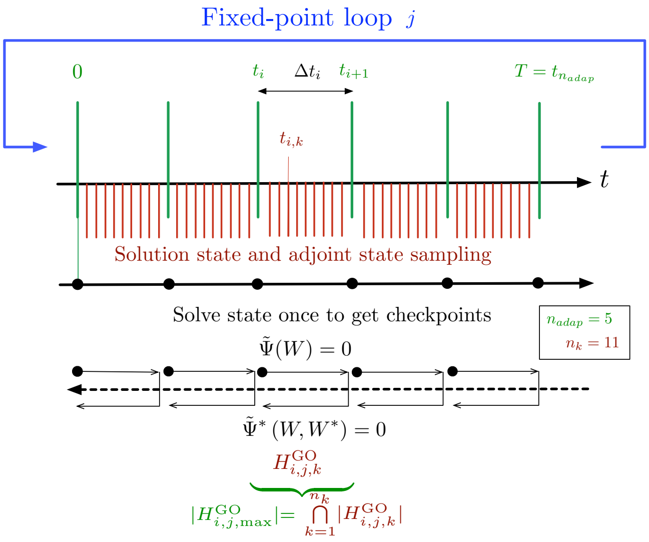

Fixed-point loop j
0
t_i
Δt_i
t_{i+1}
$T = t_{n_{adap}}$
$t_{i,k}$
t
Solution state and adjoint state sampling
Solve state once to get checkpoints
$n_{adap} = 5$
$n_k = 11$
$\tilde{\Psi}(W) = 0$
$\tilde{\Psi}^*(W, W^*) = 0$
$H^{\mathrm{GO}}_{i,j,k}$
$|H^{\mathrm{GO}}_{i,j,\max}| = \bigcap_{k=1}^{n_k} |H^{\mathrm{GO}}_{i,j,k}|$

Figure 8.4: Global Fixed-Point algorithm for unsteady goal-oriented anisotropic mesh adaptation

8.7.3 Computing the goal-oriented metric

The optimal goal-oriented metric is a function of the adjoint state, the adjoint state gradient, the state time derivative Hessian and the Euler fluxes Hessians. In practice, these continuous states are approximated by the discrete states and derivative recovery is applied to get gradients and Hessians.

The discrete adjoint state W_h^* is taken to represent the adjoint state W^*. The gradient of the adjoint state ∇W^* is replaced by $\nabla_R W_h^*$ and the Hessian of each component of the flux vector $H(\mathcal{F}_i(W))$ is obtained from $H_R(\mathcal{F}_i(W_h))$. ∇_R (resp. H_R) stands for the operator that recovers numerically the first (resp. second) order derivatives of an initial piecewise linear solution field. In this thesis, the recovery method is based on the $\mathbf{L}^2$-projection formula. Its formulation along with some comparisons to other methods is available in [Alauzet 2010a].

8.8 Numerical Experiments

The adaptation algorithms described in this thesis have been implemented the CFD code `Wolf`. As regards meshing, goal-oriented mesh adaptation requires to update the surface mesh of Γ on which the functional is observed. This standpoint is needed in order to ensure a valid coupling between the volume mesh and the surface mesh. This implies to consider a more complex re-meshing phase. In our case, a local remeshing strategy has been considered. We use `Yams` [Frey 2001] for the adaptation in 2D and `Feflo.a` [Loseille 2010b] in 3D.

8.8.1 2D Blast wave propagation

We first apply the goal-oriented adaptive strategy to the example presented in Section 8.4.3. It consists in a 2D blast in a geometry representing a city. We recall that the cost function of interest j is the quadratic deviation of the ambient pressure on target surface S (see Figure 8.1):

$$j(W) = \int_0^T \int_S \frac{1}{2}(p(\mathbf{x},t) - p_{air})^2 \mathrm{d}S\mathrm{d}t.$$

The simulation time frame is split into 30 time sub-intervals, *i.e.,* 30 different adapted meshes are used to perform the simulation. Hence, 30 checkpoints are stored for the backward computation of the unsteady adjoint.

The resulting adjoint-based anisotropic adapted meshes are shown in Figure 8.5. The corresponding density isolines are depicted in Figure 8.6. These adapted meshes indubitably illustrate that, thanks to the unsteady adjoint, the mesh adaptation only focuses on shock waves that impact the observation region and ignores other area of the flow field. Therefore, waves traveling toward the observation are accurately captured whereas the rest of the flow is poorly computed. We also observe that once waves go beyond the target surface, the mesh is no more refined even if they continue traveling throughout the computational domain. Indeed, they do not impact anymore the functional.

It is then quite interesting to compare the Hessian-based approach of [Alauzet 2011] with our adjoint-based method. This comparison can be done by considering Figures 8.6(top, right) and 8.8. It demonstrates how the adjoint defines an optimal distribution of the degrees of freedom for the specific functional, while it

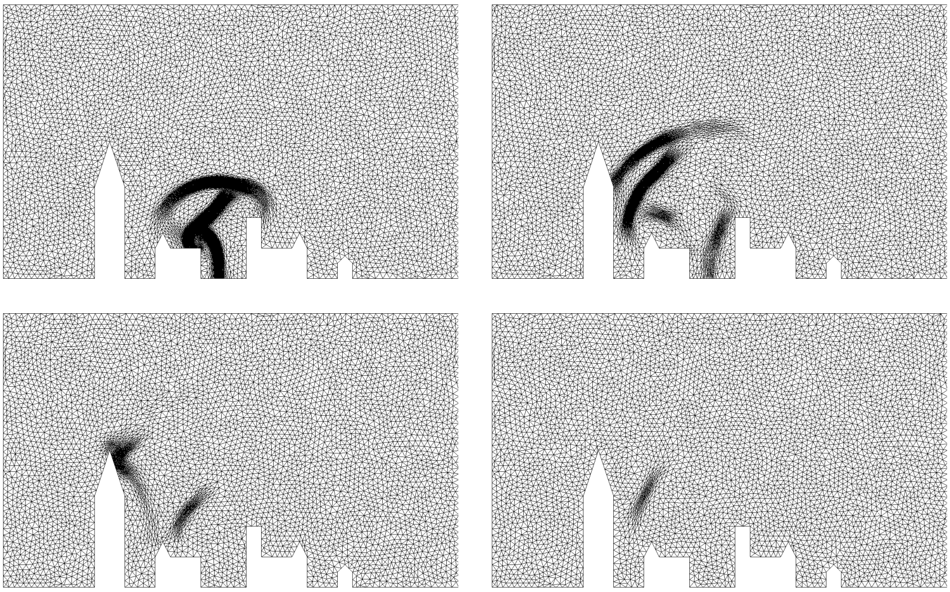


Figure 8.5: 2D city blast adjoint-based adapted meshes evolution. From top to bottom and left to right, adapted meshes corresponding to sub-intervals 8, 15, 22 and 29 at a-dimensional time 1.2, 2.25, 3.3 and 4.35.

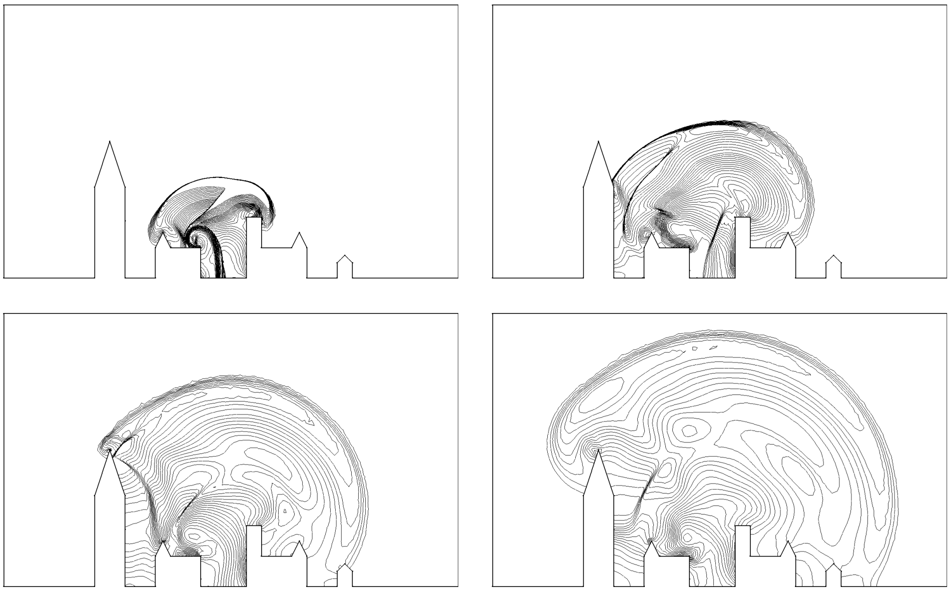


Figure 8.6: 2D city blast adjoint-based adaptive solution state evolution. From top to bottom and left to right, density iso-lines corresponding to the end of sub-intervals 8, 15, 22 and 29 at a-dimensional time 1.2, 2.25, 3.3 and 4.35.

is clear that in this context the Hessian-based approach gives a non-optimal result for the evaluation of the functional but capture accurately the whole flow.

In conclusion, if an output functional of interest is provided then the reduction of the simulation number of degrees of freedom can be even more improved by considering a goal-oriented analysis instead of an Hessian-based methodology.

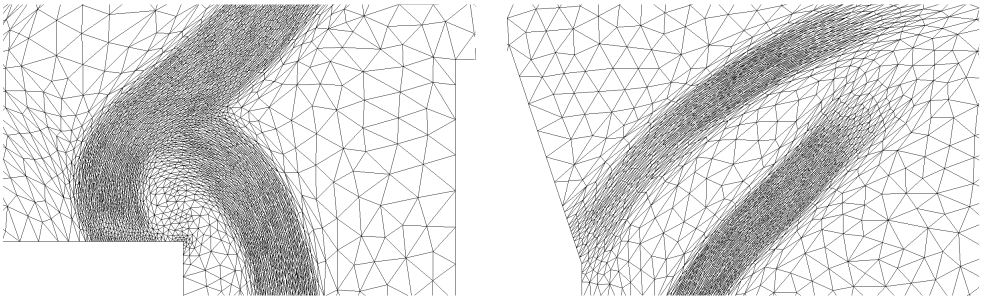


Figure 8.7: Zoom in on adjoint-based adapted meshes corresponding to sub-intervals 8 (left) and 15 (right). Mesh anisotropy is clearly visible.

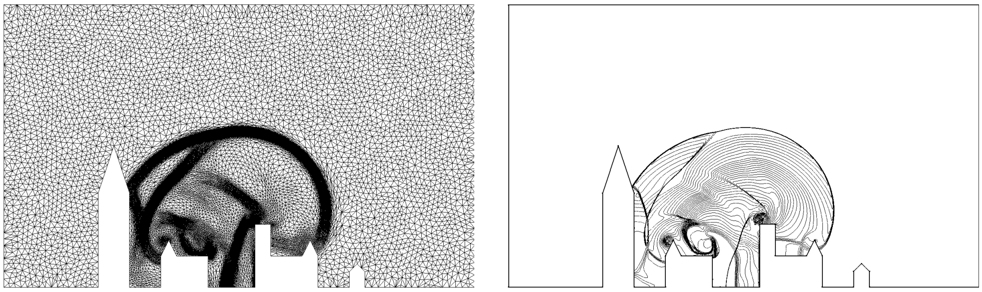


Figure 8.8: Adapted mesh and corresponding density iso-lines for sub-interval 15 at a-dimensional time 2.25 obtained with the Hessian-based method of [Alauzet 2011].

8.8.2 2D Acoustic wave propagation

Another typical example of pressure deviation propagating over long distances is linear acoustic. Linear acoustic waves usually refer either to a transient wave of bounded duration or to a periodic vibration. An important context in the study of these different kinds of waves is when we are interested only in the effect of the wave on a microphone occupying a very small part of the region affected by the pressure perturbation. Further simplifying, we can be interested in a single scalar measure of this effect, for instance the total energy E_{tot} received by the sensor during a given time interval. If the pressure perturbation is emitted at a very long distance in an open and complex spatial domain, the numerical simulation of this phenomenon can be extremely computer intensive, if not impossible.

We consider a sound source located at the center-bottom of a rectangular domain. An acoustic source defined by $f = (0, 0, 0, r)$, where:

$$r = -Ae^{-Bln(2)\left[x^2+y^2\right]}Ampl \cdot cos(2\pi f),$$

is added as a source term on the right-hand side of the unsteady compressible Euler Equations (8.9). Constants are $A = 0.01$, $B = 256$, $Ampl = 2.5$ and $f = 2$ is the waves frequency. We analyze the sound signal emitted by this source on a microphone M located at the center-top of the domain.

Goal-oriented mesh adaptation considers cost function:

$$j(W) = \int_0^T \int_M \frac{1}{2}(p(\mathbf{x},t) - p_0)^2 \,\mathrm{d}M\mathrm{d}t\,.$$

The simulation is split into 40 sub-intervals (thus 40 checkpoints are used) and nnn fixed-point iteration to converge the mesh adaptation problem. As expected, mesh adaptation reduces as much as possible mesh fineness in parts of the computational domain where accuracy loss does not influence the quality of sound prediction on the micro. This is illustrated in Figure 8.9. Moreover, small perturbations of the waves on the uniform mesh are no more visible on the adapted anisotropic mesh, as shown in Figure 8.10. Therefore, the solution computed on the adapted meshes is perfectly smooth. In Figure 8.11 a zoom on the density field with associated uniform and anisotropic adapted mesh can be seen.

Accuracy study An accuracy analysis of the proposed method are carried out on integrand $k(t)$ on the micro M:

$$k(t) = \int_M (p - p_0)\mathrm{d}M\,,$$

for different sizes of uniform and adapted meshes. $k(t)$ is plotted for three uniform meshes in Figure 8.12. It illustrates that, for a rather coarse mesh of about $60,000$ nodes, a small perturbation appears at the entrance of the micro. This perturbation diminishes with finer meshes. This spurious behavior is completely avoided by adaptive computations. Figure 8.13 supports our affirmation and, as expected, our functional has a better prediction with adapted meshes.

In the case of higher frequencies, the choice of the mesh is even more crucial for a good prediction of the waves. Indeed, an increase of the frequency induces a drastic diminution of the spatial mesh size to preserve the computation accuracy:

$$\Delta x = \frac{\lambda}{n} = \frac{c}{n\,f}\,,$$

where λ is the wave length, c the sound of speed and n the number of elements per wave length (usually $n \geq 5$). This leads to a huge increase in CPU's, therefore acoustic wave propagation in real-world applications remains a great numerical challenge. We simulate a higher frequency wave propagation, with $f = xxx$, on a uniform mesh composed of 117 000 vertices and we compare it to a goal-oriented adaptive simulation. The obtained integrand $k(t)$ for both simulations is depicted in Figure 8.14 which points out that in that case the choice of the mesh is even more crucial. Indeed, the proposed adaptive method is able to capture on the micro all the frequencies of the generated waves whereas a poor behavior is obtained with the uniform mesh.

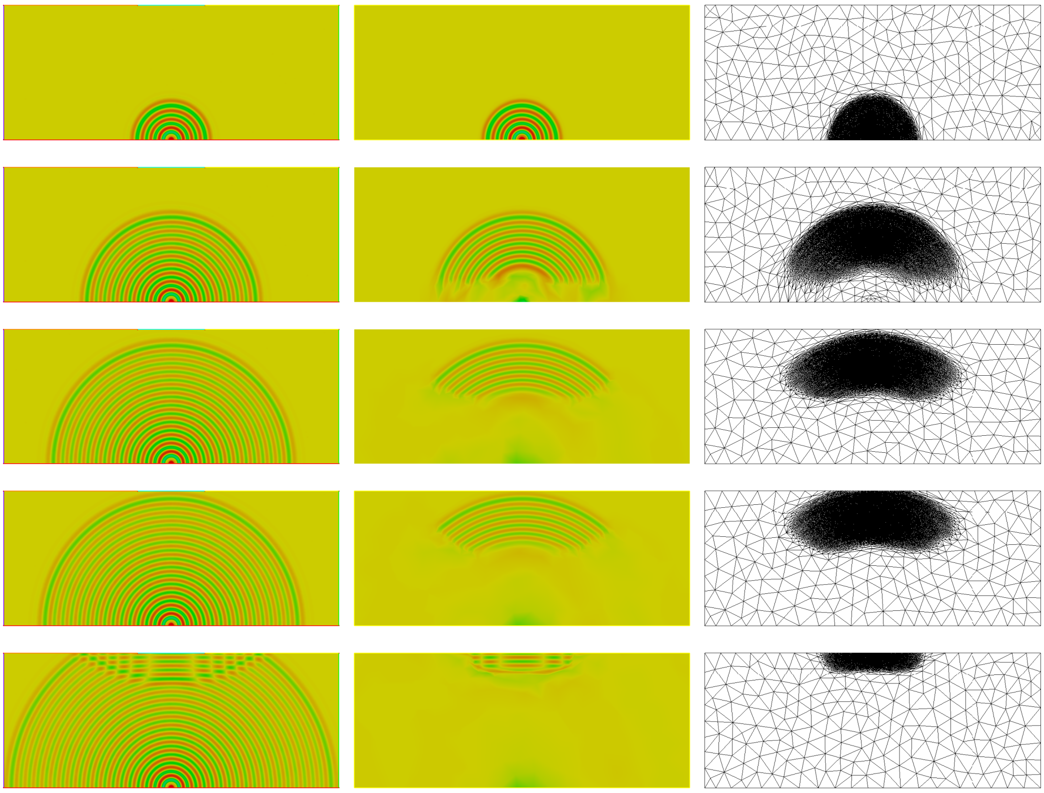


Figure 8.9: Propagation of acoustic waves: density field evolving in time on a uniform mesh (left) and on adapted meshes (middle and right).

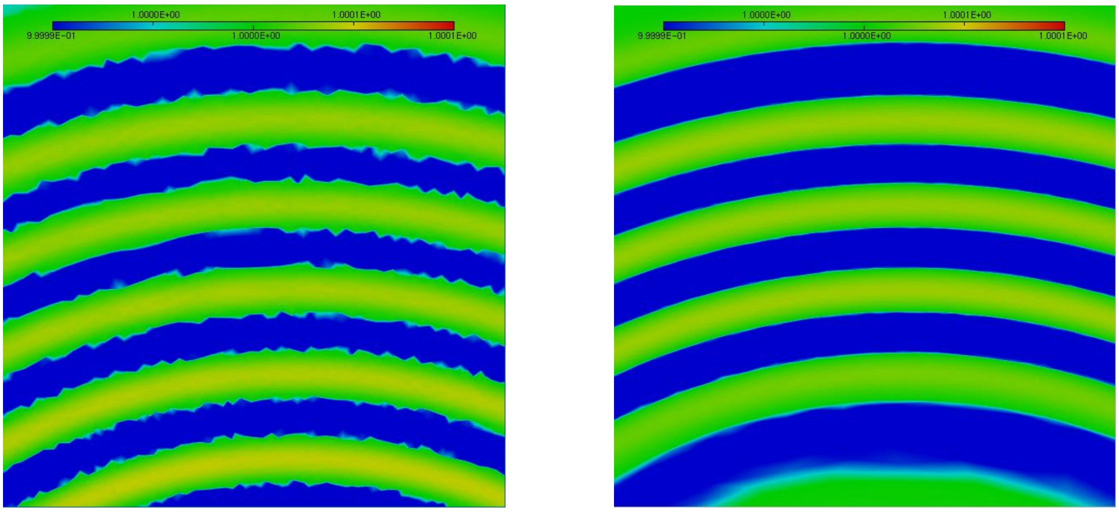


Figure 8.10: Acoustic wave represented on uniform mesh (left) and on adapted mesh (right)

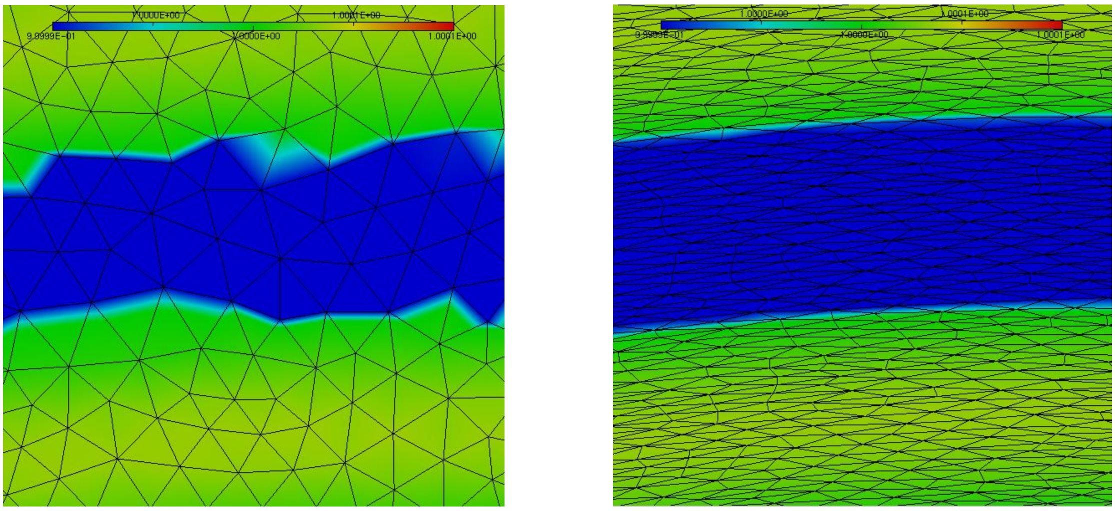


Figure 8.11: Mesh visualization of uniform mesh (left) and adapted mesh (right)

Convergence analysis Now, a numerical convergence order analysis is done on the previous simulation with a wave frequency $f = 2$ in order to verify mathematical results of Section 8.6. For a mesh of N vertices, we denote by u_N and u_{exact} the approximate and exact solutions, respectively. The following relation holds:

$$u_N(\mathbf{x}, t) \;=\; u_{exact}(\mathbf{x}, t) \;+\; N^{-\frac{\alpha}{d}} u_1(\mathbf{x}, t) \;+\; o(N^{-\frac{\alpha}{d}})$$

for d spatial dimension, α the convergence parameter to be found and u_1 the first normalized error term. Since α cannot be directly determined, an estimation is done

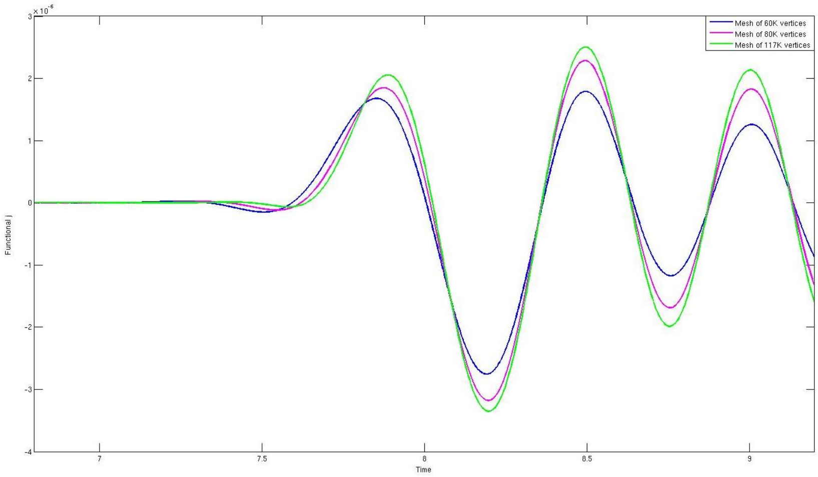


Figure 8.12: Functional time integrand $k(t)$ on uniform meshes of size 60K, 80K and 117K vertices for wave propagation with frequency $f = 2$.

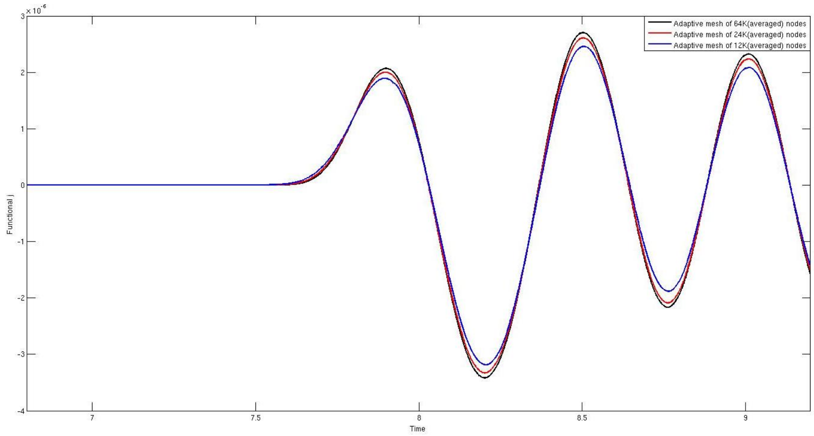


Figure 8.13: Functional time integrand $k(t)$ on series of adaptive simulations with average meshes size of 12K, 24K and 64K vertices for wave propagation with frequency $f = 2$.

on three meshes of different sizes: N_1, N_2 and N_3. Suppose u_{N_1}, u_{N_2} and u_{N_3} the corresponding numerical solutions, then we seek for α such that :

$$\frac{1 - \frac{N_2}{N_1}^{-\frac{\alpha}{d}}}{1 - \frac{N_3}{N_1}^{-\frac{\alpha}{n}}} \approx \frac{u_{N_1} - u_{N_2}}{u_{N_1} - u_{N_3}} \tag{8.41}$$

with dimension $d = 2$ in our case. Furthermore, we make the assumption that N_1 represent the higher number of vertices and N_3 the lowest one.

We solve Equation (8.41) for both uniform and adapted meshes simulations presented above. The considered values for u_{N_i}, with $i = 1, 2, 3$, are the first maximal value of integrand $k(t)$ which are plotted in Figures 8.12 and 8.13 for uniform meshes and for adaptive simulations, respectively. Table 8.1 summarizes the data collection:

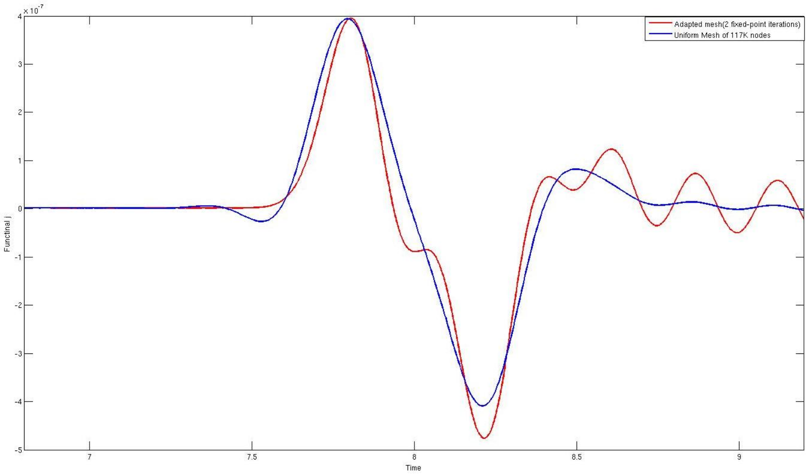


Figure 8.14: Functional time integrand $k(t)$ for high frequency wave propagation with $f = xxx$ for adaptive (red curve) and uniform mesh (blue curve) simulations.

the convergence order is found to be 0.6 for uniform meshes and 1.98 for the adaptive simulations.

Mesh	**First observed maximum**	**Convergence order**
Uniform mesh 60K nodes	$1.67578e-06$	**0.6**
Uniform mesh 80K nodes	$1.84838e-06$	
Uniform mesh 117K nodes	$2.05461e-06$	
		1.98
Adapted meshes 12K (average) nodes	$1.89061e-06$	
Adapted meshes 24K (average) nodes	$1.99895e-06$	
Adapted meshes 64K (average) nodes	$2.06722e-06$	

Table 8.1: Mesh convergence for the time-dependent pressure deviation on observation area.

8.8.3 3D Blast wave propagation

Finally, the last example consider a purely three-dimensional blast problem in a complex geometry representing a city. The city size is $85\,m \times 70\,m \times 70\,m$. In this simulation, shock waves interact with each other and are reflected by buildings. The city geometry is the same as in [Alauzet 2007]. Initially, the ambient air is at rest: $W_{air} = (1, 0, 0, 2.5)$. A "blast-like" initialization is considered inside a half-sphere of radius $2.5\,m$ around position $(42, 53, 0)$: $W_{blast} = (10, 0, 0, 250)$. Cost function j is again the quadratic deviation from ambient pressure on target surface Γ which is composed of one building for simulation 1 or two buildings for simulation 2, see Figure 8.15:

$$j(W) = \int_0^T \int_\Gamma \frac{1}{2}(p(\mathbf{x}, t) - p_{air})^2 \mathrm{d}\Gamma \mathrm{d}t.$$

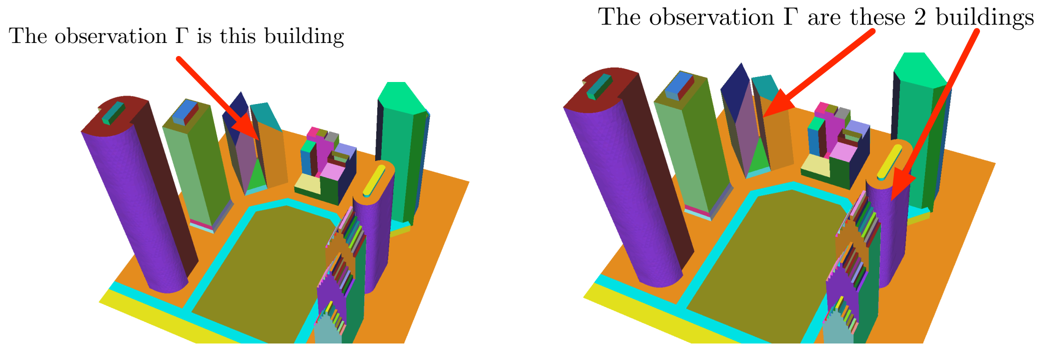

The observation Γ is this building
The observation Γ are these 2 buildings

Figure 8.15: 3D City test case geometry and location of target surface Γ composed of one building for simulation 1 (left) or two buildings for simulation 2 (right).

The simulation time frame is split into 40 time sub-intervals, *i.e.,* 40 different adapted meshes are used to perform the simulation. 6 fixed-point iterations are performed to converge the mesh adaptation problem. For each sub-interval 16 samples of the solution and adjoint states are considered to build the goal-oriented metric. Both simulations consider the same space-time complexity equal to 1.2 million.

The resulting adjoint-based anisotropic adapted meshes (surface and volume) for both simulations at sub-interval 10, 15 and 20 are shown in Figures 8.16 and 8.17. It is very interesting to see that we are not restricted to just one target surface. Despite the complexity and the unpredictable behavior of the physical phenomena with a large number of shock waves interactions with the geometry, the goal-oriented fixed point mesh adaptation algorithm was automatically able to capture all shock waves which impact the targeted buildings and to set appropriate weights for the refinement of these physical phenomena. Other waves are neglected thus leading to a drastic reduction of the mesh size.

To illustrate this point, we provide meshes size of simulation 1 and 2 for sub-intervals 1, 5, 10 and 20 in Table 8.2 and 8.3, respectively. Where nv, nt, nf are the number of vertices, tetrahedra and triangles and h the mesh size. If for sub-interval 1 one million vertices is needed, only forty thousands vertices are used for sub-interval 20. The required mesh size has been reduced by a factor 25 between the first and the twentieth sub-interval. In average, almost 200 000 vertices are required to perform both adaptive computations with a maximal accuracy between 1 and 5 cm. These numbers have to be compared with uniform meshes characteristics given in Table 8.4 where dozens of millions vertices are required to reach an accuracy of 30 cm.

As regards the amount of anisotropy achieved for these simulations, mesh anisotropy can be quantified by two different indicators: the anisotropic ratios and quotients. Deriving these quantities for an element relies on the fact that there always exists a unique metric tensor $\mathcal{M}_K$ for which this element is unit. Once $\mathcal{M}_K$ is computed, the anisotropic ratios and quotients associated with element K are simply given by

$$\text{ratio} = \sqrt{\frac{\min_i \lambda_i}{\max_i \lambda_i}} = \frac{\max_i h_i}{\min_i h_i} \quad \text{and} \quad \text{quo} = \frac{\max_i h_i^3}{h_1 h_2 h_3}.$$

where $(\lambda_i)_{i=1,2,3}$ are the eigenvalues of $\mathcal{M}_K$ and $(h_i)_{i=1,2,3}$ are the corresponding sizes. The anisotropic ratio stands for the maximum elongation of a tetrahedron by comparing two principal directions. For both simulations an average anisotropic ratio between 7 and 16 is achieved. The anisotropic quotient represents the overall anisotropic ratio of a tetrahedron taking into account all the possible directions, we get a mean anisotropic quotient between 40 and 200. This quotient can be considered as a measure of the overall gain in three dimensions of *an anisotropic adapted mesh* as compared to *an isotropic adapted one*, here almost 100. The gain is of course even greater when compared to a uniform mesh.

Iteration	nv	nt	nf	min h	ratio avg. (max)	quotient avg. (max)
1	1 058 084	6 177 061	79 862	1. cm	9 (62)	63 (2242)
10	249 620	1 427 956	34 318	1.2 cm	16 (94)	197 (5136)
15	72 432	392 970	24 678	2.1 cm	12 (76)	119 (3789)
20	40 297	205 855	21 980	4.8 cm	7 (64)	39 (3547)
Avg.	188 357	1 074 777	28 811			

Table 8.2: Mesh characteristics for simulation 1 of 3D blast wave calculation.

Iteration	nv	nt	nf	min h	ratio avg. (max)	quotient avg. (max)
1	1 051 805	6 139 926	82 858	1. cm	9 (53)	67 (1829)
10	233 116	1 327 503	36 376	1.2 cm	16 (84)	203 (4083)
15	77 446	421 024	26 008	2.7 cm	12 (85)	124 (3654)
20	45 500	235 383	23 072	6 cm	7 (48)	51 (1673)
Avg.	195 355	1 113 492	31 110			

Table 8.3: Mesh characteristics for simulation 2 of 3D blast wave calculation.

Uniform mesh	nv	nt	nf	average h	ratio	quotient
1	10 294 136	60 826 207	862 264	43 cm	2 (12)	3 (116)
2	33 352 859	198 163 572	2 135 032	29 cm	2 (14)	3 (165)

Table 8.4: Uniform mesh characteristics for the 3D city geometry.

8.9 Conclusion

We have designed a new mesh adaptation algorithm which prescribes the spatial mesh of an unsteady simulation as the optimum of a goal-oriented error analysis. This method specifies both mesh density and mesh anisotropy by variational calculus. Accounting for unsteadiness is applied in a time-implicit mesh-solution coupling

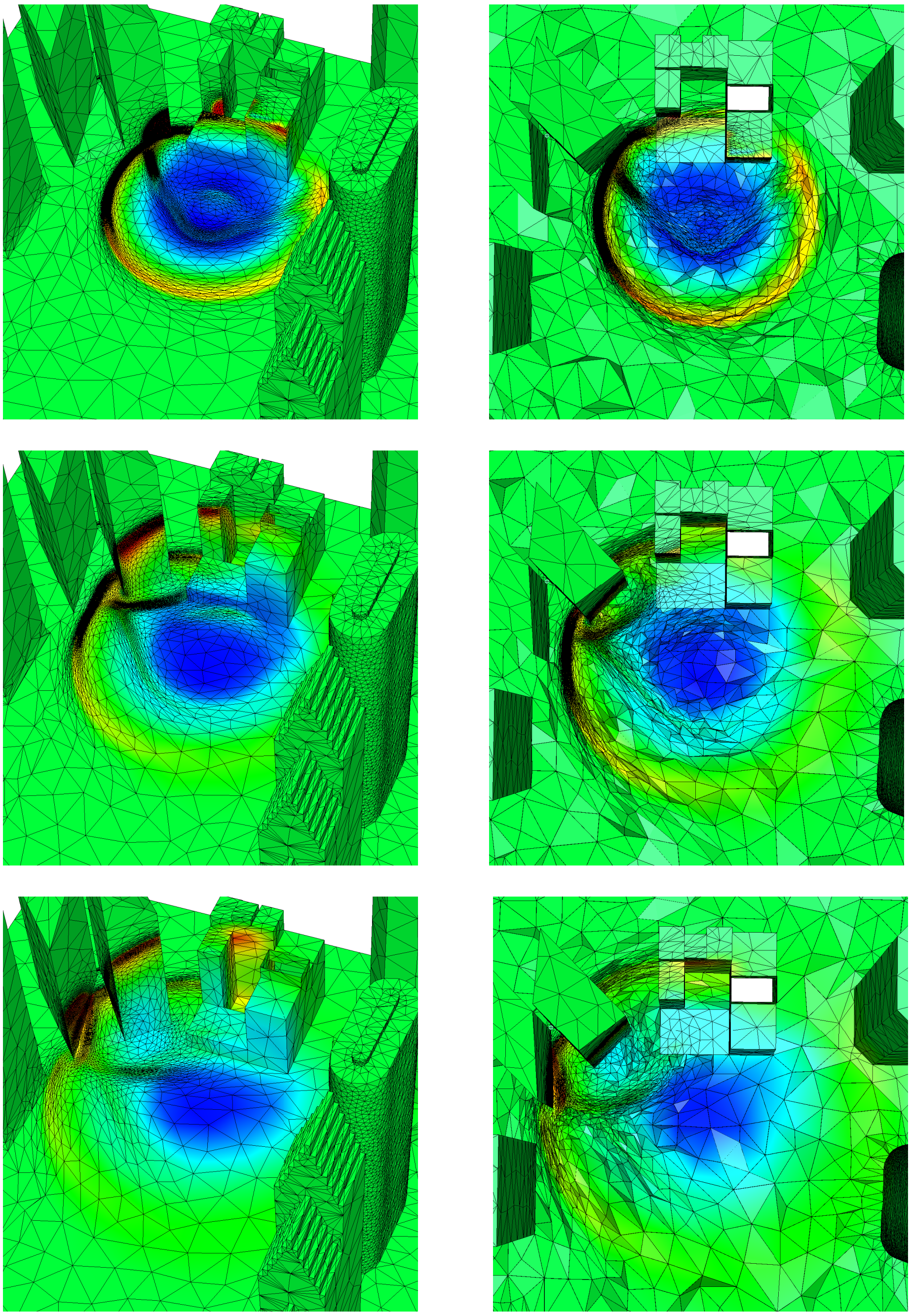


Figure 8.16: 3D Blast wave propagation: simulation 1. Adjoint-based anisotropic adapted surface (left) and volume (right) meshes at sub-interval 10, 15 and 20 and corresponding solution density at a-dimensioned time 5, 7.5 and 10.

which needs a non-linear iteration, the fixed point. In contrast to the Hessian-based fixed-point of [Alauzet 2007, Guégan 2010] which iterates on each sub-interval, the new iteration covers the whole time interval, including forward steps for evaluating the state and backward ones for the adjoint. This algorithm has been successful ap-

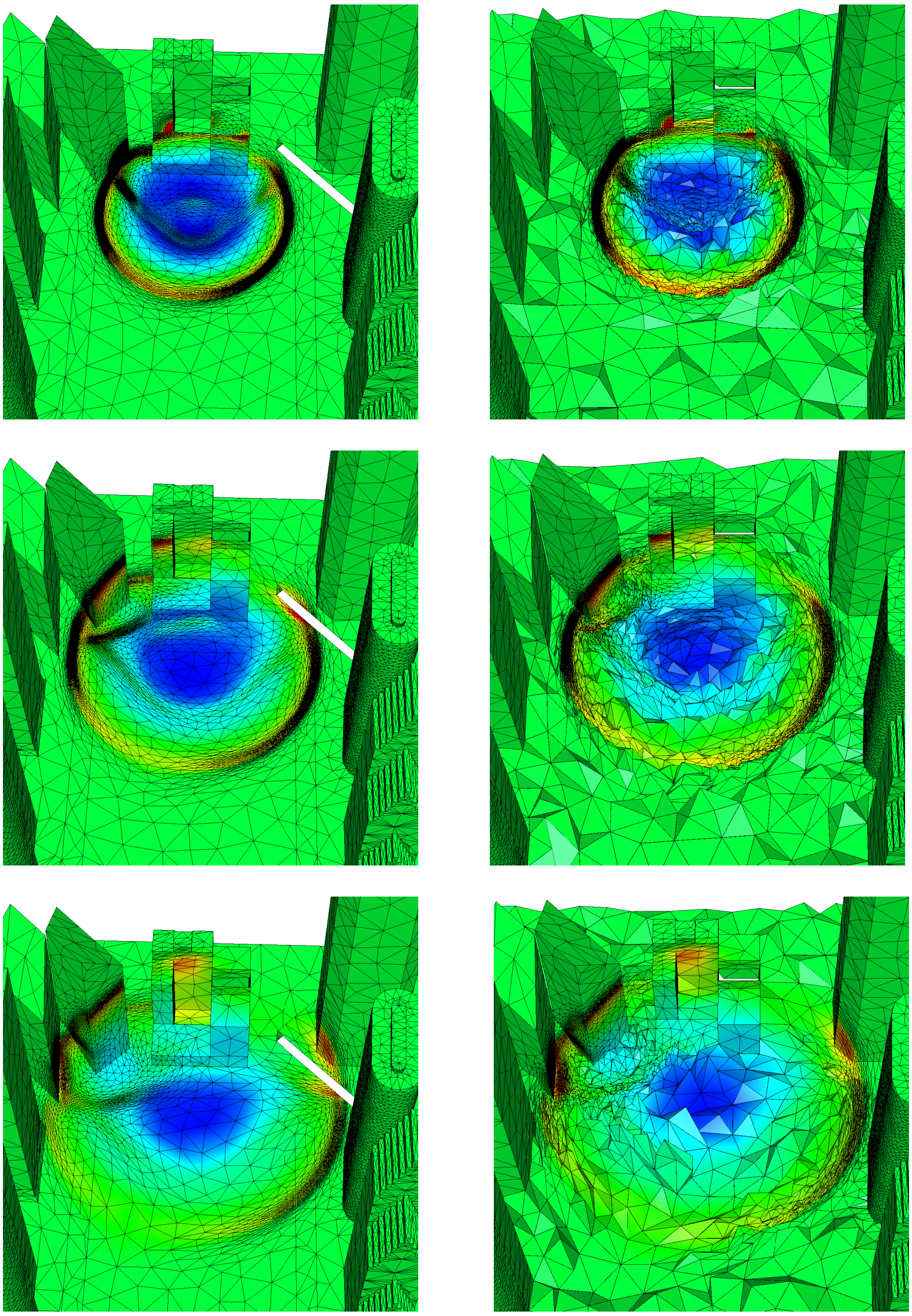


Figure 8.17: 3D Blast wave propagation: simulation 2. Adjoint-based anisotropic adapted surface (left) and volume (right) meshes at sub-interval 10, 15 and 20 and corresponding solution density at a-dimensioned time 5, 7.5 and 10.

plied to 2D and 3D blast wave Euler test cases and to the calculation of a 2D acoustic wave. Results demonstrate the favorable behavior expected from an adjoint-based method, which gives an automatic selection of the mesh regions necessary for the target output.

Several important issues for fully space-time computation have been addressed in this chapter. Among them, the strategies for choosing the splitting in time sub-intervals and the accurate integration of time errors in the mesh adaptation process have been proposed, together with a more general formulation of the mesh optimization problem.

Time discretization error is not considered in this study. Solving this question is not so important for the type of calculation that are shown here, but can be of paramount impact in many other cases, in particular when implicit time advancing is considered. In a future work, we plan to consider a space-time error analysis in the context of the proposed method.

Chapter 9

Goal-oriented anisotropic mesh adaptation method for unsteady Navier-Stokes flows

Contents

9.1 Introduction

This chapter is concerned with the mesh adaptation for unsteady Navier-Stokes flows. The case of unsteady Euler flow was addressed in previous Chapter 8.
The extension of a mesh adaptation method from Euler models to Navier-Stokes models is a difficult and challenging field of investigation. A review of the main problematics and contributions can be found in [Fidkowski 2011]. Adressing the Navier-Stokes model, it introduces new problems which can be identified as belonging to three families:

1. high mesh stretching or anisotropy,
2. error criteria,
3. turbulence.

- First, the most difficult issue today is the adaptative generation of highly stretched meshes as should be used in high-Reynolds flow. Indeed, stretching ratio (typically ratio between largest side to smallest altitude in a tetrahedron) as high as $10^4 - 10^6$ can be necessary for computing a viscous flow past an aircraft. Further, stretching needs in principle to be automatised in a mesh adaptation loop, either from scratch, or from an existing mesh which has to be improved. Three main families of algorithms have been considered in the literature. Mesh deformation for Navier-Stokes dates back at least to Gnoffo's work [Gnoffo 1983] and was illustrated in 2D for planetary entry of a spatial probe. Metric-based mesh regeneration is now much more frequently used, but the maximal stretching in that latter case is still unsufficient.

- Second, an important and difficut issue is the building of sufficiently good error sensors. Indeed, most error estimates are accurate only if the considered mesh is fine enough. This constitutes an existencial problem for algorihms the goal of which is to find a sufficiently good mesh. Most truncation-based error estimates do not allow a specification of the anisotropy of the adapted mesh. In Venditti *et al.* [Venditti 2003b], the truncation-based error is complemented by a Hessian-based interpolation error (typically of Mach field) for specifying the mesh anisotropy (see also [Park 2003]).

- Lastly, a quite open problem concerns the many difficulties arising when a turbulent flow is considered. RANS modeling allows to transform the initial problem into a steady numerical model, (see [Mavriplis 1990, Johnson 1995, Venditti 2003b, Park 2010, Nielsen 2004]) but the realistic unsteady chaotic case has not been much adressed.

In this chapter, we start from the novel anisotropic goal-oriented method, built in Chapter 8 and extend it to the unsteady (laminar) Navier-Stokes model. We show first how the Navier-Stokes estimates developed in Chapter 6 can be introduced into an error sensor. The minimisation of an integral of the error sensor perfectly specifies an optimal anisotropic mesh as it is shown in Section 9.2.2. Then a few preliminary numerical examples are proposed in Section 9.4 for the flow around a NACA airfoil.

The notations used here are those introduced in Chapter 8.

9.2 Adaptation criterion for unsteady Navier-Stokes model

9.2.1 Error bound

Consider the following compact variational formulation of Navier-Stokes System in the functional space $\mathcal{Q} = H^1\{[0,T]\,;V\}$:

$$
\begin{aligned}
&\text{Find } W \in \mathcal{Q} \text{ such that } \forall \varphi \in \mathcal{Q}, \quad (\Psi(W)\,,\,\varphi) = 0\\
\text{with} \quad (\Psi(W)\,,\,\varphi) &= \int_0^T \int_\Omega \varphi W_t \,\mathrm{d}\Omega\,\mathrm{d}t\\
&+ \int_0^T \int_\Omega \varphi\,(\nabla\cdot\mathcal{F}(W) + \nabla\cdot\mathcal{V}(W))\,\mathrm{d}\Omega\,\mathrm{d}t\\
&- \int_0^T \int_\Gamma \varphi\,\hat{\mathcal{F}}(W).\mathbf{n}\,\mathrm{d}\Gamma\,\mathrm{d}t\,. \qquad (9.1)
\end{aligned}
$$

with associated semi-discrete formulation:

$$
\begin{aligned}
&\text{Find } W_h \in \mathcal{Q}_h \text{ such that } \forall \varphi_h \in \mathcal{Q}_h, \quad (\Psi_h(W_h)\,,\,\varphi_h) = 0,\\
\text{with:} \quad (\Psi_h(W_h)\,,\,\varphi) &= \int_0^T \int_\Omega \varphi\;\Pi_h W_{h,t}\,\mathrm{d}\Omega\,\mathrm{d}t\\
&+ \int_0^T \int_\Omega \varphi\,(\nabla\cdot\mathcal{F}_h(W_h) + \nabla\cdot\mathcal{V}_h(W_h))\,\mathrm{d}\Omega\,\mathrm{d}t\\
&- \int_0^T \int_\Gamma \varphi\hat{\mathcal{F}}_h(W_h).\mathbf{n}\,\mathrm{d}\Gamma\,\mathrm{d}t \qquad (9.2)
\end{aligned}
$$

Refering to as one of the "goals" of this thesis, we focus on errors committed on a target functional $j(W) = (g, W)$. Following again the idea of Loseille *et al.* in [Loseille 2010a] these error can be *a priori* estimated as :

$$
\delta j \approx ((\Psi_h - \Psi)(W), W^*)\ . \qquad (9.3)
$$

Replacing operators Ψ and Ψ_h by their expressions (9.1) and respectively (9.2), discarding the error commited when imposing the initial condition and neglecting the effect of the numerical viscosity term we finally get:

$$
\begin{aligned}
\delta j \;\approx\; & \int_0^T \int_\Omega W^*\,\big(W - \Pi_h W\big)_t\,\mathrm{d}\Omega\,\mathrm{d}t + \int_0^T \int_\Omega W^*\,\nabla.\big(\mathcal{F}(W) - \Pi_h\mathcal{F}(W)\big)\,\mathrm{d}\Omega\,\mathrm{d}t\\
&+ \int_0^T \int_\Omega W^*\,\nabla.\big(\mathcal{V}(W) - \mathcal{V}_h(W)\big)\,\mathrm{d}\Omega\,\mathrm{d}t - \int_0^T \int_\Gamma W^*\,\big(\hat{\mathcal{B}}(W) - \hat{\mathcal{B}}_h(W))\big).\mathbf{n}\,\mathrm{d}\Gamma\mathrm{d}t\,.
\end{aligned}
$$

With $\hat{\mathcal{B}}$ we denoted the boundary terms and W^* the adjoint state.

As a next step, in the previous Euler analysis we performed an integration by parts. In the case of Navier-Stokes System the viscous part is highly non-linear and consequently more difficult to analyse and an integration by parts at this step would be useless. Nevertheless, we recall that in Chapter 6 an *a priori* analysis for this viscous part has been developed and can be applied here. Thus, after separation of Euler fluxes and viscous fluxes, since they are threated in a different manner, an integration by parts is applied to Euler fluxes and the a priori estimator (based globally on summation of interpolation errors weighted by spectral radius of adjoint state Hessians) to the latest viscous part:

$$\begin{aligned}\delta j \;\approx\;& \int_0^T\int_\Omega W^*\big(W-\Pi_h W\big)_t\,\mathrm{d}\Omega\,\mathrm{d}t - \int_0^T\int_\Omega \nabla W^*\big(\mathcal{F}(W)-\Pi_h\mathcal{F}(W)\big)\,\mathrm{d}\Omega\,\mathrm{d}t\\ &+ \int_0^T \delta E\;\mathrm{d}t - \int_0^T\int_\Gamma W^*\big(\bar{\mathcal{F}}(W)-\Pi_h\bar{\mathcal{F}}(W))\big).\mathbf{n}\,\mathrm{d}\Omega\,\mathrm{d}t\,.\end{aligned}$$

With $\delta E = \int_\Omega W^*\,\nabla.\big(\mathcal{V}(W)-\mathcal{V}_h(W)\big)\mathrm{d}\Omega\,\mathrm{d}t$ we denote the error terms contribution from viscous flux. Symbol $\bar{\mathcal{F}} = \hat{\mathcal{B}} - \mathcal{F}$ represents the boundary terms, the contribution of which is neglected in the sequel.

According to Estimate (**??**) at the end of Chapter 6 for the viscous terms we have:

$$\begin{aligned}\delta E \;\approx\;& \sum_{i=1}^{3}\left(\frac{27}{2}\int_\Omega \mu|\rho(H(W^*_{i+1}))||u_i-\Pi_h u_i|\mathrm{d}\Omega + \sum_{j=1}^{3}\frac{3}{2}\int_\Omega \mu|\rho(H(W^*_{i+1}))||u_j-\Pi_h u_j|\mathrm{d}\Omega\right)\\ &+ \frac{9}{2}\int_\Omega |\lambda||\rho(H(W^*_5))|\;|\Pi_h T - T|\mathrm{d}\Omega + \frac{9}{2}\int_\Omega \mu\;\sum_{i=1}^{3}|u_i||\rho(H(W^*_5))||u_i-\Pi_h u_i|\;\mathrm{d}\Omega\\ &+ \sum_{i=1}^{3}\frac{9}{2}\int_\Omega \mu\;|u_i||\rho(H(W^*_5))||\mathbf{u}-\Pi_h\mathbf{u}|\;\mathrm{d}\Omega + 3\;\int_\Omega \mu\;|\mathbf{u}||\rho(H(W^*_5))||\mathbf{u}-\Pi_h\mathbf{u}|\;\mathrm{d}\Omega\\ &- \frac{1}{3}\left[\int_\Omega \mu\;\nabla W^*_5\left((\mathbf{u}-\Pi_h\mathbf{u})\nabla\cdot\mathbf{u}\right)\mathrm{d}\Omega - \int_\Omega \mu\;\nabla W^*_5\cdot\nabla\mathbf{u}.(\mathbf{u}-\Pi_h\mathbf{u})\mathrm{d}\Omega\right].\end{aligned} \tag{9.4}$$

We observe that this estimate of δj is expressed in terms of:

- interpolation error on time derivatives W_t weighted by continuous function W^*
- interpolation errors of the Euler fluxes $\mathcal{F}$ weighted by continuous function ∇W^*
- and from viscous fluxes we add the sum of :
  - interpolation errors on velocity vector $\mathbf{u} = (u_1, u_2, u_3)$ components weighted by spectral radius of adjoint Hessians $\rho(H(W^*))$; we underline here that for non-linear terms from the last line of viscous matrix we obtain products of interpolation errors with cross velocity and adjoint Hessians: $|u_i||\rho(H(W^*))||u_j - \Pi_h u_j|$,
  - interpolation error on temperature T weighted by spectral radius of adjoint Hessians $\rho(H(W^*))$,
  - and interpolation error on velocity vector weighted by the gradient of the fifth component of adjoint state multiplied by a scalar value (obtained as the product of constants with the divergence of the velocity vector).

Error bound with a safety principle. The integrands in Error Estimation (9.4) contain positive and negative parts which can compensate for some particular meshes. In our strategy, we prefer to not rely on these parasitic effects and to slightly over-estimate the error. To this end, all integrands are bounded by their absolute values:

$$\begin{aligned}(g, W_h - W) \;\leq\; & \int_0^T \int_\Omega |W^*| \, |\big(W - \Pi_h W\big)_t| \, \mathrm{d}\Omega \, \mathrm{d}t \\ + \; & \int_0^T \int_\Omega |\nabla W^*| \, |\mathcal{F}(W) - \Pi_h \mathcal{F}(W)| \, \mathrm{d}\Omega \, \mathrm{d}t + \int_0^T |\delta E| \; \mathrm{d}t \\ + \; & \int_0^T \int_\Gamma |W^*| \, |(\bar{\mathcal{F}}(W) - \Pi_h \bar{\mathcal{F}}(W)).\mathbf{n}| \, \mathrm{d}\Omega \, \mathrm{d}t \,. \end{aligned} \tag{9.5}$$

9.2.2 Optimal error model

As for Euler flows, we work in the continuous framework, such that, the continuously discret meshes are modeled by Riemannian metric spaces.
Working in this framework enables us to write Estimate (9.5) in a continuous form:

$$\begin{aligned}|(g, W_h - W)| \approx \mathbf{E}(\mathbf{M}) \;=\; & \int_0^T \int_\Omega |W^*| \; |\big(W - \pi_{\mathcal{M}} W\big)_t| \, \mathrm{d}\Omega \, \mathrm{d}t \\ + \; & \int_0^T \int_\Omega |\nabla W^*| \; |\mathcal{F}(W) - \pi_{\mathcal{M}} \mathcal{F}(W)| \, \mathrm{d}\Omega \, \mathrm{d}t + \int_0^T |\delta E_{\mathcal{M}}| \; \mathrm{d}t \\ + \; & \int_0^T \int_\Gamma |W^*| \; |(\bar{\mathcal{F}}(W) - \pi_{\mathcal{M}} \bar{\mathcal{F}}(W)).\mathbf{n}| \, \mathrm{d}\Gamma \, \mathrm{d}t, \end{aligned} \tag{9.6}$$

where, we recall, $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x} \in \Omega}$ is a continuous mesh defined by a Riemannian metric space and $\pi_{\mathcal{M}}$ is the continuous linear interpolate defined above.

Once again we discard the error terms from the boundary surface mesh, and we refer to [Loseille 2010a] for a discussion of the influence of it.

Then, we can write the simplified error model as follows:

$$\mathbf{E}(\mathbf{M}) \;=\; \int_0^T \int_\Omega \mathrm{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x}, t) \, \mathbf{H}(\mathbf{x}, t) \, \mathcal{M}^{-\frac{1}{2}}(\mathbf{x}, t)\right) \, \mathrm{d}\Omega \, \mathrm{d}t$$

$$\text{with} \quad \mathbf{H}(\mathbf{x}, t) = \sum_{j=1}^{5} \left([\Delta t]_j(\mathbf{x}, t) + [\Delta x]_j(\mathbf{x}, t) + [\Delta y]_j(\mathbf{x}, t) + [\Delta z]_j(\mathbf{x}, t)\right) + \Delta E, \tag{9.7}$$

in which

$$\begin{aligned}
[\Delta t]_j\,(\mathbf{x},t) &= \big|W_j^*(\mathbf{x},t)\big| \cdot \big|H(W_{j,t})(\mathbf{x},t)\big|, \\
[\Delta x]_j\,(\mathbf{x},t) &= \left|\frac{\partial W_j^*}{\partial x}(\mathbf{x},t)\right| \cdot \big|H(\mathcal{F}_1(W_j))(\mathbf{x},t)\big|, \\
[\Delta y]_j\,(\mathbf{x},t) &= \left|\frac{\partial W_j^*}{\partial y}(\mathbf{x},t)\right| \cdot \big|H(\mathcal{F}_2(W_j))(\mathbf{x},t)\big|, \\
[\Delta z]_j\,(\mathbf{x},t) &= \left|\frac{\partial W_j^*}{\partial z}(\mathbf{x},t)\right| \cdot \big|H(\mathcal{F}_3(W_j))(\mathbf{x},t)\big|.
\end{aligned}$$

Here, W_j^* denotes the j^{th} component of the adjoint vector W^*, $H(\mathcal{F}_i(W_j))$ the Hessian of the j^{th} component of the vector $\mathcal{F}_i(W)$, and $H(W_{j,t})$ the Hessian of the j^{th} component of the time derivative of the conservative vector W.

The additional term ΔE represents the contribution from the viscous part. From Relation 9.4 it writes:

$$\begin{aligned}
\Delta E \;\approx\; & \sum_{i=1}^{3}\left(\frac{27}{2}|\mu\rho(H(W_{i+1}^*))|\big|H(u_i)\big| + \sum_{j=1}^{3}\frac{3}{2}|\mu\rho(H(W_{i+1}^*))|\big|H(u_j)\big|\right) \\
+ \;& \frac{9}{2}|\lambda\rho(H(W_5^*))|\big|H(T)\big| + \frac{9}{2}\sum_{i=1}^{3}|\mu\ u_i\ \rho(H(W_5^*))|\big|H(u_i)\big| \\
+ \;& \sum_{i=1}^{3}\frac{9}{2}|\mu\ u_i\ \rho(H(W_5^*))|\big|H(\mathbf{u})\big| + 3\ |\mu\ \mathbf{u}\ \rho(H(W_5^*))|\big|H(\mathbf{u})\big| \\
- \;& \frac{1}{3}\left[|\mu\ \nabla W_5^*\nabla\cdot\mathbf{u}|\big|H(\mathbf{u})\big| - |\mu\ \nabla W_5^*\cdot\nabla\mathbf{u}|\big|H(\mathbf{u})\big|\right].
\end{aligned}$$

We underline in the above expression that $\big|H(*)\big|$ are Hessians approaching the interpolation errors, and $\rho(H(W^*)$ are scalar values weights (computed in the numerical code, and given as inputs to metric computation). We recall that T is the temperature and $\mathbf{u}$ the velocity vector of components $u_i,\ i=1,2,3$.

The mesh optimization problem writes:

$$\text{Find } \mathbf{M}_{opt} = \text{Argmin}_{\mathbf{M}}\,\mathbf{E}(\mathbf{M}), \tag{9.8}$$

under the constraint of bounded mesh fineness:

$$\mathcal{C}_{st}(\mathbf{M}) = N_{st}, \tag{9.9}$$

where N_{st} is a specified total number of nodes. Since we consider an unsteady problem, the space-time (st) complexity used to compute the solution takes into account the time discretization. The above constraint then imposes the total number of nodes in the time integral, that is:

$$\mathcal{C}_{st}(\mathbf{M}) = \int_0^T \tau(t)^{-1}\left(\int_\Omega d_{\mathcal{M}}(\mathbf{x},t)\mathrm{d}\mathbf{x}\right)\,\mathrm{d}t \tag{9.10}$$

where $\tau(t)$ is the time step used at time t of interval $[0, T]$.

The discussion of Chapter 8 regarding the definition of an optimal space-time metric holds here too. Nevertheless, we restrict to the practical choice for the goal-oriented metric, that is, the one defined in Section 8.7.1 that we recall here:

- the Hessian metric for time sub-interval i is based on a control of the temporal error in $\mathbf{L}^\infty$ norm:

$$\mathbf{H}^i_{\mathbf{L}^\infty}(\mathbf{x}) = \Delta t_i \max_{t \in [t_i, t_{i+1}]} \mathbf{H}(\mathbf{x}, t) = \Delta t_i \, \mathbf{H}^i_{\max}(\mathbf{x}) \,,$$

- function $\tau : t \to \tau(t)$ is constant and equal to 1,
- all sub-intervals have the same time length Δt.

The optimal goal-oriented metric $\mathbf{M}_{go} = \{\mathbf{M}^i_{go}\}_{i=1,..,n_{adap}}$ then simplifies to:

$$\mathcal{M}^i_{go}(\mathbf{x}) = N_{st}^{\frac{2}{3}} \left(\sum_{i=1}^{n_{adap}} (\int_\Omega (\det \mathbf{H}^i_{\max}(\mathbf{x}))^{\frac{1}{5}} \mathrm{d}\mathbf{x}) \right)^{-\frac{2}{3}} (\Delta t)^{\frac{1}{3}} \left(\det \mathbf{H}^i_{\max}(\mathbf{x}) \right)^{-\frac{1}{5}} \mathbf{H}^i_{\max}(\mathbf{x}) \,.$$

9.3 Numerics

The optimality system (state, adjoint, optimal metric) is discretized:

- for state and adjoint, using the mixed element-volume method described in the first chapter of this thesis,
- for optimal metrics, using the methods as for Euler case, Chapter 8.

The global transient fixed-point algorithm of Chapter 8 is applied.

9.4 Applications

We present in the sequel some preliminary results of our mesh adaptation algorithm for a laminar unsteady Navier-Stokes flow. This is a standard 2D benchmark for testing unsteady laminar flow solvers [D. Guiraud 1988, Pulliam 1989]. A viscous flow around a NACA0012 airfoil, at Mach number 0.1 and an angle of attack of 30 degrees is considered. Physically, it represents a high incidence gust inducing the separation of the flow on the upper side of the airfoil. The Reynolds number is 1000. The simulation is performed starting from an unstructured triangular mesh of 7379 vertices (see Figure 9.1) for a physical time of 3 seconds.

The target functional is the drag force observed on the target surface Γ which in this case is the profile:

$$j(W) = \int_0^T \int_\Gamma (p - p_\infty) n_x \mathrm{d}\Gamma \mathrm{d}t. \tag{9.11}$$

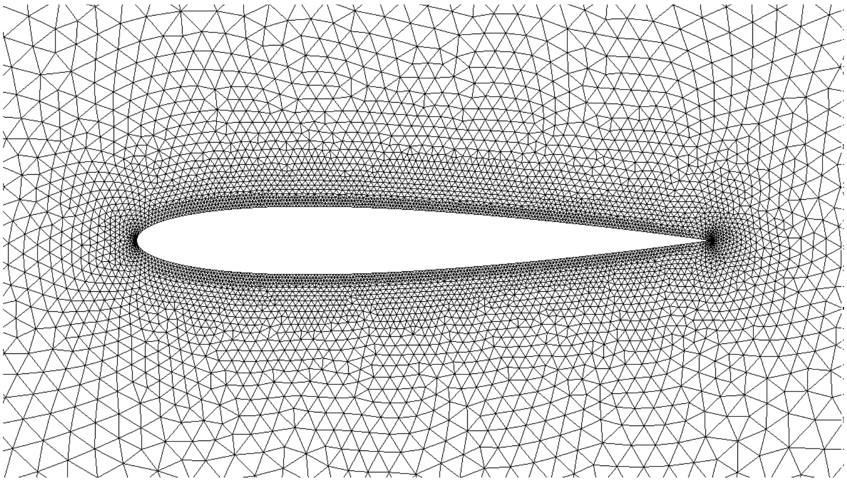


Figure 9.1: Snapshot on initial spatial mesh (7379 vertices). Zoom near the NACA.

The algorithm of computation is identical to the previously described one, based on checkpointing. An explicit scheme of Runge-Kutta is used for time advancing and second order *HLLC* spatial solver. Figure 9.2 (resp. 9.3) illustrates the density (resp. Mach number) evolution through time, we observe a process of vortex formation due to the high angle of attack.

An illustration of the unsteady adjoint solution for Navier-Stokes flow is shown in Figure 9.4 at several instantaneous times.
After a number of two adaptation loops, we can see the remeshing proposed by our algorithm in Figure 9.5. We observe that the mesh show anisotropy only on viscous boundary layer region and along the separating viscous layer.

9.5 Concluding remarks

The proposed application is still evolving. A first conclusion concerns the robustness of the new platform: the adjoint state is stably solved. It produces the expected weight on our mesh metric. Preliminary results are qualitatively as good as for the Euler case.

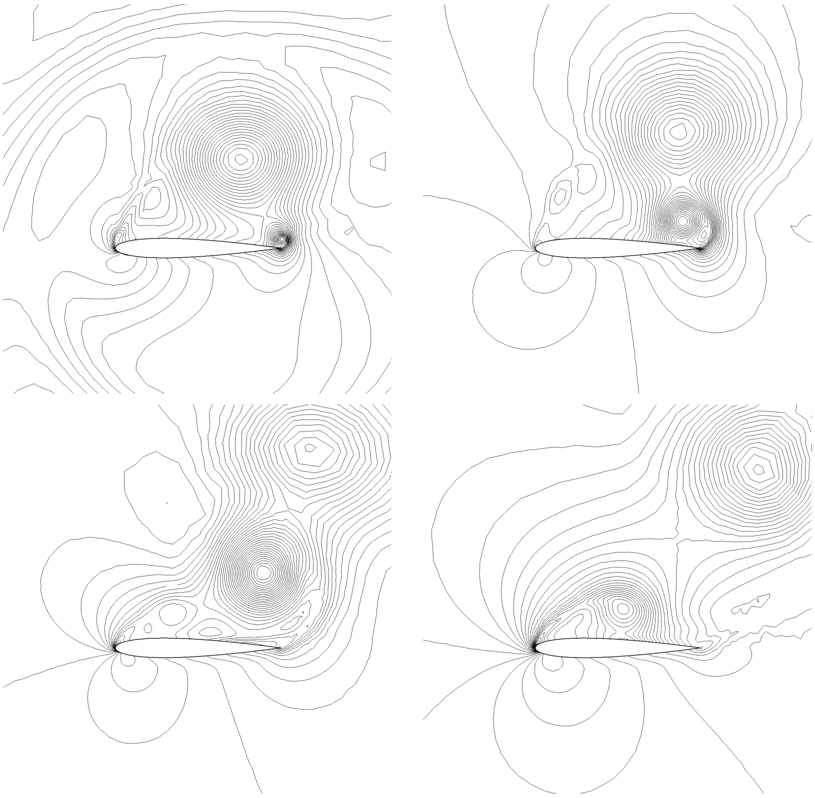


Figure 9.2: Density field evolution of the viscous Navier-Stokes flow around the NACA0012 airfoil, Mach=0.1 and angle of attack $\alpha = 30$.

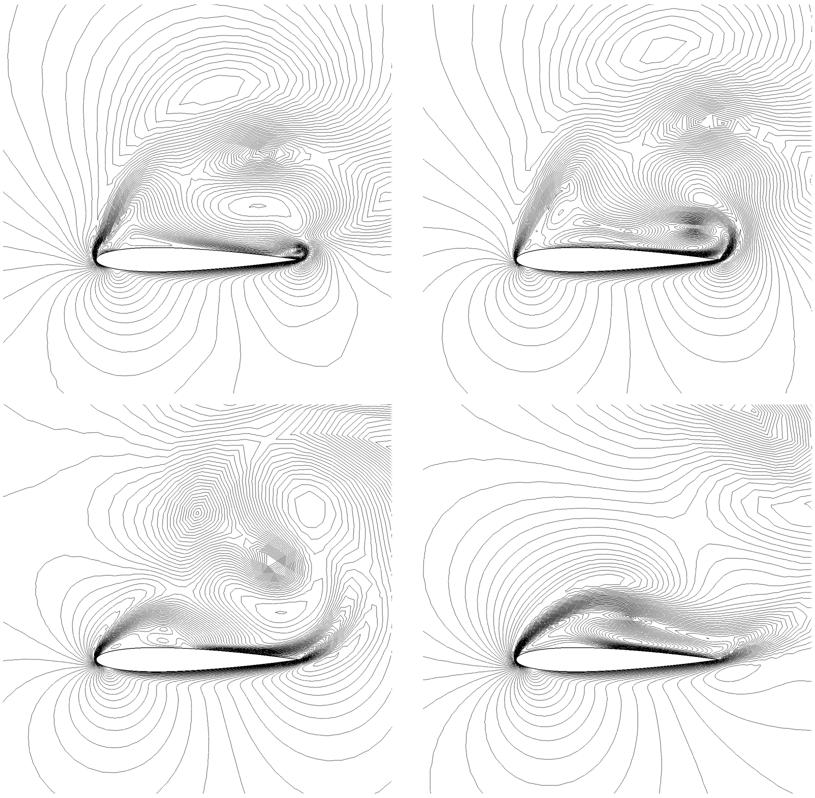


Figure 9.3: Mach field evolution of the viscous Navier-Stokes flow around the NACA0012 airfoil, Mach=0.1 and angle of attack $\alpha = 30$.

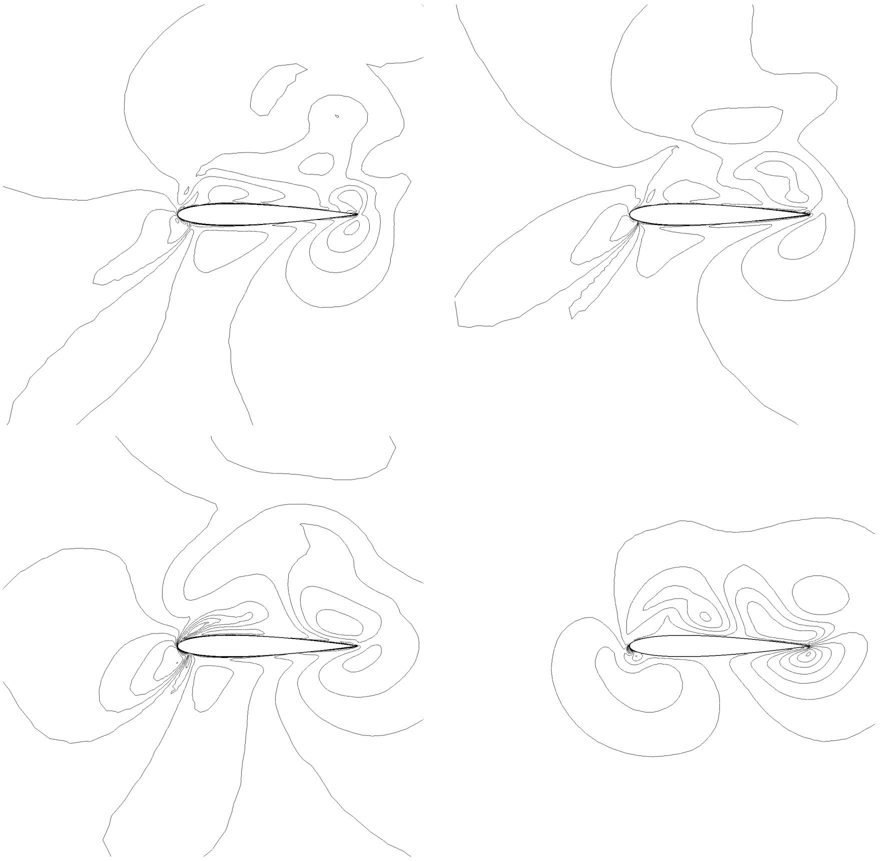


Figure 9.4: From left to right and top to bottom, snapshot of the adjoint-density isolines at a-dimensional time 0.12, 1.5, 2. and 2.88 .

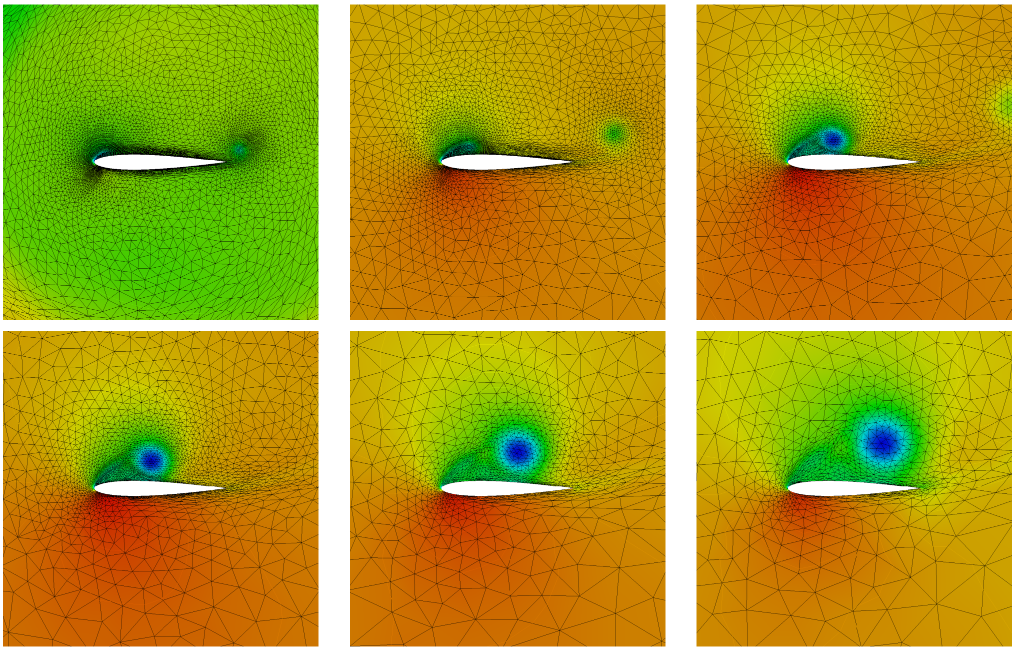


Figure 9.5: Plot of instantaneous density on the adapted meshes: evolution from top to bottom, left to right at a-dimensional time 0.12, 1.5, 2. and 2.88 .

Chapter 10

Conclusion and Perspectives

We have presented in this thesis our contribution to two complementary domains: turbulence modelling and mesh adaptation for unsteady flows based on error estimates.

They are complementary from the standpoint of a final combination of both. Coupling turbulent flows computation with mesh adaptation based on output of observation or not, still is a big challenge in CFD. But we can expect from it an important increment in turbulence prediction. We have approached these problematics by developping both sides and by studying error estimates for viscous flows, a bridge between the two domains.

Contribution on turbulent flows The progress of methods for turbulent flows simulation generally seems a slow process. However, during this decade important engineering advances are obtained by combining existing methods and technologies, like statistical modelling, Large Eddy Simulation, unstructured meshes, and Parallel Computing. In this thesis we have addressed both industrial and academic point of view of turbulent flows modelisation. From a more academic point of view we have proposed, studied, and evaluated in Chapter 2 and 3 an improvement of the numerical scheme by adding a mass matrix formulation in the sense that the time derivative of Navier-Stokes system is approximated by a formulation closer to finite-element. Next, the major contribution in turbulent flows is the developpement in Chapter 3 of the hybrid scheme with a dynamic SGS closure and a protection zone to avoid mismatches near the boundary. For both contributions, our benchmark have been the flow around a circular cylinder, a widely used geometry, but a challenging test case at the higher Reynolds numbers considered.

Contribution on error correction Chapters 4, 5 and 6 have been dedicated to error correction methods. Managing errors is of paramount importance for numericists since they deal with them every day. In this thesis we have considered both the a posteriori and a priori errors. For the first one we have clarified an existing analysis, remarked that adjoint state is not always compulsory, and demonstrated on a steady problem the interest in avoiding the adjoint. Regarding the a priori error estimation, we have contributed on the developpement of a truncation error analysis for elliptic problems, showing how it can be bounded by an interpolation error weighted by a second derivative of a discrete Lagrange multiplier. This theoretical result lead us to an a priori estimator for viscous fluxes, of paramount importance for our apporach in anisotropic mesh adaptation.

Contribution on output-based anisotropic mesh adaptation Numerous papers in the literature adressed the topic of output-based error estimate and it still remains a modern and complex subject of research. Among them, very few authors applied these output-based estimators to anisotropic mesh adaptation, and probably a even small part to unsteady computations. We have proved in this thesis that such adapted meshes can be generated for an accurately defined error estimator, producing in fine an accurate goal-oriented metric. The novel algorithm of global transient fixed-point has been also a part of the success of this work. Several issues for integrating time in our algorithm have been proposed too, in a more theoretical viewpoint. The diversification of our test cases (blast problems and acoustics) suggests the globality of application of our methods.

This thesis is a small step in mesh adaptation and turbulent flows simulations. It is also an open gate for further investigations and thus future thesis. A list of these new perpectives is given now:

Perspectives:

Turbulent flows simulation A lot of questions remains open for turbulent flow simulation. Regarding the direct sequels of this thesis, deeper analysis of the hybrid model can be considered. For example, we have presented the hybrid model as a simplified version of a "double field" version, based on the Non-Linear Disturbance Equation principle. This idea can be pushed further either in an actual double-field numerical model, or more cleverly, into a formulation in which the RANS component has less influence on small scales, by using again the Variational Multiscale principle. This can allow addressing in a novel way the difficult problem of the geometrical interface between RANS and LES. Another important issue is the evaluation of an adjoint for a LES or hybrid model. The automatic differentiation tools developed in the Tropics team can be of good help for this developement. Target applications are of course error correction or mesh adaptation.

Error estimation The methods of analysis for a priori error estimates have been extended from first-order hyperbolic models to elliptic and parabolic second order ones. Many other models of physics can be addressed with our estimates, for example MHD, etc. A posteriori estimates are closer to the error since they do not restrict to the main asymptotic terms as a priori ones. New estimates could combine both a posteriori estimates and anisotropic a priori ones.

Mesh adaptation We have tackled and proposed some preliminary results of our algorithm applied to laminar viscous flows, with, as theoretical support, the a priori error estimator from Chapter 6. Further investigations on laminar flows has to be done, in particular in the steady cases, before application to statistical turbulence models (like one equation Spalart-Allmaras model) and then to hybrid models. Secondly, regarding the transient fixed point, the mesh is frozen in time sub-interval. After the advances in mesh deformation obtained during the thesis of Geraldine Olivier [Olivier 2011], it would be interesting to develop a transient fixed point in which the mesh deforms in the time sub-intervals. Third, we have proposed several issues to include time in the metric definition, either when we consider a single time interval or severals sub-intervals. Investigation of this metrics can be considered in the future.

Bibliography

[A. Belme 2010] B. Koobus A. Belme A. Dervieux, S. Wornom and M-V. Salvetti. *Application of Hybrid and VMS-LES turbulent models to aerodynamic simulations.* In Proceedings of the ICAS 27th International Council of the Aeronautical Sciences, 2010. (Cited on pages 46 and 48.)

[A. Bueno-Orovio 2007] F. Palacios A. Bueno-Orovio C. Castro and E. Zuazua. *Continuous adjoint approach for the Spalart-Allmaras model in aerodynamic optimization.* AIAA Journal, vol. 45, no. 9, pages 2125–2139, 2007. (Cited on page 91.)

[Abalakin 2001] I. Abalakin, V. Bobkov, A. Dervieux, T. Kozubskaya and V. Shiryaev. *Study of high accuracy order schemes and non-reflecting boundary conditions for noise propagation problems.* Liapunov Institute report, project 99-02, available on :http://www-sop.inria.fr/tropics/Alain.Dervieux/liapunov.html, 2001. (Cited on page 27.)

[Abalakin 2002a] I. Abalakin, A. Dervieux and T. Kozubskaya. *A vertex centered high order MUSCL scheme applying to linearised Euler acoustics.* Research report RR-4459, INRIA, 2002. (Cited on pages 25 and 27.)

[Abalakin 2002b] I. Abalakin, A. Dervieux and T. Kozubskaya. *Computational Study of Mathematical Models for Noise DNS.* AIAA paper, pages 2002–2585, 2002. (Cited on page 27.)

[Abalakin 2002c] I. Abalakin, A. Dervieux and T. Kozubskaya. *High Accuracy Study of Mathematical Models for DNS of Noise around Steady Mean Flow.* In D.E. Zeitoun, J. Periaux, J.A. Desideri and M. Marini, editeurs, Proc. of West East High Speed Flow Field Conference 2002. CIMNE, Barcelona, 2002. (Cited on page 27.)

[Abalakin 2004] I. Abalakin, A. Dervieux and T. Kozubskaya. *On accuracy of noise direct calculation based on Euler model.* Int. J. Acoustics, vol. 3, no. 2, pages 157–180, 2004. (Cited on page 27.)

[Absil 2008] P.-A. Absil, R. Mahony and R. Sepulchre. Optimization algorithms on matrix manifolds. Princeton University Press, Princeton, NJ, 2008. (Cited on page 152.)

[Achenbach 1968] E. Achenbach. *Distribution of local pressure and skin friction around a circular cylinder in cross-flow up to* $Re = 5 \cdot 10^6$. J. Fluid Mech, vol. 34, no. 4, pages 625–639, 1968. (Cited on pages 65, 70 and 71.)

[Alauzet 2003a] F. Alauzet and P.J. Frey. *Estimateur d'erreur géométrique et métrique anisotropes pour l'adaptation de maillage. Partie I : aspects* *théoriques.* RR-4759, INRIA, March 2003. (in French). (Cited on pages 136 and 137.)

[Alauzet 2003b] F. Alauzet, P.L. George, B. Mohammadi, P.J. Frey and H. Borouchaki. *Transient fixed point based unstructured mesh adaptation.* Int. J. Numer. Meth. Fluids, vol. 43, no. 6-7, pages 729–745, 2003. (Cited on page 167.)

[Alauzet 2007] F. Alauzet, P.J. Frey, P.-L. George and B. Mohammadi. *3D transient fixed point mesh adaptation for time-dependent problems: Application to CFD simulations.* J. Comp. Phys., vol. 222, pages 592–623, 2007. (Cited on pages 143, 158, 162, 167, 169, 177 and 180.)

[Alauzet 2009] F. Alauzet and A. Loseille. *On the use of space filling curves for parallel anisotropic mesh adaptation.* In IMR 18, pages 337–357. Springer, 2009. (Cited on page 169.)

[Alauzet 2010a] F. Alauzet and A. Loseille. *High Order Sonic Boom Modeling by Adaptive Methods.* J. Comput. Phys., vol. 229, pages 561–593, 2010. (Cited on pages 146, 161 and 170.)

[Alauzet 2010b] F. Alauzet and M. Mehrenberger. *P1-conservative solution interpolation on unstructured triangular meshes.* Int. J. Numer. Meth. Eng., 2010. Submitted. (Cited on pages 166 and 168.)

[Alauzet 2011] F. Alauzet and G. Olivier. *Extension of Metric-Based Anisotropic Mesh Adaptation to Time-Dependent Problems Involving Moving Geometries.* In 49th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2011-0896, Orlando, FL, USA, Jan 2011. (Cited on pages 143, 158, 162, 167, 168, 170 and 172.)

[Anderson 1997] K. Anderson and V. Venkatakrishnan. *Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation.* AIAA 35'th Aerospace Sciences Meeting Exhibit, 1997. (Cited on page 91.)

[Apel 2007] T. Apel and S. Nicaise. *A posteriori error estimations of a SUPG method for anisotropic diffusion-convection-reaction problems.* C. R. Acad. Sci. Paris, vol. I:345, pages 657–662, 2007. (Cited on page 110.)

[Arian 1997] E. Arian and D. Salas. *Admitting the Inadmissible: Adjoint Formulation for Incomplete Cost Functionals in Aerodynamic Optimization.* Research Report 97-69, ICASE, december 1997. (Cited on page 91.)

[Arsigny 2006] V. Arsigny, P. Fillard, X. Pennec and N. Ayache. *Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors.* Magn. Reson. Med., vol. 56, no. 2, pages 411–421, 2006. (Cited on pages 137, 138 and 152.)

[Babuska 2001] I. Babuska and T. Strouboulis. The finite element method and its reliability. Clarendo Press, Oxford, 2001. (Cited on page 130.)

[Baines 1994] M.J. Baines. Moving finite elements. Oxford University Press, Inc., New York, NY, 1994. (Cited on page 143.)

[Barth 1994] T.J. Barth. *Aspects of Unstructured Grids and Finite Volume Solvers for the Euler and Navier-Stokes Equations.* Von Karman Institute Lecture Notes, 1994. (Cited on page 16.)

[Becker 1996a] R. Becker and R. Rannacher. *A feed-back approach to error control in finite element methods: basic analysis and examples.* East-West J. Numer. Math., vol. 4, 1996. (Cited on pages 4 and 110.)

[Becker 1996b] R. Becker and R. Rannacher. *A feed-back approach to error control in finite element methods: basic analysis and examples.* East-West J. Numer. Math., vol. 4, pages 237–264, 1996. (Cited on page 143.)

[Becker 2001] R. Becker and R. Rannacher. *An optimal control approach to a posteriori error estimation in finite element methods.* Acta Numerica, vol. 19, pages 1–102, 2001. (Cited on pages 4 and 83.)

[Belhamadia 2004] Y. Belhamadia, A. Fortin and E. Chamberland. *Three-dimensional anisotropic mesh adaptation for phase change problems.* J. Comp. Phys., vol. 201, pages 753–770, 2004. (Cited on page 142.)

[Belme 2011] A. Belme, M-V. Salvetti S. Wornom B. Koobus and A. Dervieux. *Assessment of VMS-LES and hybrid RANS VMS-LES models.* In Proceedings of the AIMETA XX Congresso di Associazione Italiana di Meccanica Teoretica e Applicata, 2011. (Cited on page 48.)

[Belytschko 1978] T. Belytschko and R. Mullen. *On Dispersive Properties of Finite Element Solutions.* Modern Problems in Elastic Wave Propagation, pages 67–82, 1978. (Cited on page 28.)

[Berger 1989] M. Berger and P. Colella P. *Local Adaptive Mesh Refinement for Shock Hydrodynamics.* J. Comp. Phys., vol. 82, no. 1, pages 67–84, 1989. (Cited on page 143.)

[Berger 2003] M. Berger. A panoramic view of riemannian geometry. Springer Verlag, Berlin, 2003. (Cited on page 152.)

[Blazek 2001] J. Blazek. Computational fluid dynamics: Principles and applications. Elsevier Science, Oxford England, 2001. (Cited on page 43.)

[Borouchaki 1996] H. Borouchaki and P.-L. George. *Maillage de surfaces paramétriques. Partie I: Aspects théoriques.* Research Rapport RR-2928, INRIA Rocquencourt, 1996. (in french). (Cited on page 130.)

[Bottasso 2004] C.L. Bottasso. *Anisotropic mesh adaption by metric-driven optimization.* Int. J. Numer. Meth. Engng, vol. 60, pages 597–639, 2004. (Cited on page 142.)

[C. Farhat 2005] A. Kajasekharan C. Farhat and B. Koobus. *A dynamic variational multiscale method for large eddy simulations on unstructured meshes*. Comp. Methods in Applied Mechanics and Engineering, pages 1668–1691, 2005. (Cited on pages 50 and 51.)

[Cabello 1991] J. Cabello, R. Lohner and O.-P. Jacquotte. *A variational method for the optimization of directionally stretched elements generated by the advancing front method (afm)*. In A.S. Arcilla, J.H auser, P.R. Eiseman and J.F. Thompson, editeurs, Numerical Grid Generation in Computational Field Simulation and Related Fields, Proceedings of the 3rd International Grid Conference, page 521. North Holland, 1991. (Cited on page 130.)

[Cai 1999] X.-C. Cai and M. Sarkis. *A restricted additive Schwarz preconditioner for general sparse linear systems*. SIAM Journal on Scientific Computing, vol. 21, no. 2, pages 792–797, September 1999. (Cited on page 27.)

[Camarri 1999] S. Camarri and M-V. Salvetti. *Towards the large-eddy simulation of complex engineering flows with unstructured grids*. Technical report RR-3844, INRIA, 1999. (Cited on page 49.)

[Camarri 2001] S. Camarri, M.V. Salvetti, B. Koobus and A. Dervieux. *Numerical diffusion based on high-order derivatives in MUSCL schemes for LES on unstructured grids*. In DLES-4, Direct and Large-Eddy Simulation-IV, July 18-20 2001. Twente (Holland). (Cited on page 27.)

[Camarri 2002a] S. Camarri, M.-V. Salvetti, A. Dervieux and B. Koobus. *A low diffusion MUSCL scheme for LES on unstructured grids*. Rapport technique RR-4412, INRIA, 2002. (Cited on page 27.)

[Camarri 2002b] S. Camarri, M. V. Salvetti, B. Koobus and A. Dervieux. *Large-eddy simulation of a bluff-body flow on unstructured grids*. Int. J. Num. Meth. Fluids, vol. 40, pages 1431–1460, 2002. (Cited on page 49.)

[Camarri 2004] S. Camarri, B. Koobus, M.V. Salvetti and A. Dervieux. *A low diffusion MUSCL scheme for* LES *on unstructured grids*. Computers and Fluids, vol. 33, pages 1101–1129, 2004. (Cited on page 27.)

[Camarri 2005] S. Camarri, M. V. Salvetti, B. Koobus and A. Dervieux. *Hybrid RANS/LES simulations of bluff body flow*. Wind Struct., vol. 8, pages 407–426, 2005. (Cited on pages 45, 54 and 57.)

[Carpentier 1995] R. Carpentier. *Approximation d'écoulements instationnaires. Application à des instabilités tourbillonnaires*. Thesis, University of Nice-Sophia Antipolis, 1995. (Cited on page 24.)

[Castro 2007] C. Castro and E. Zuazua. *Systematic Continuous Adjoint Approach to Viscous Aerodynamic Design on Unstructured Grids*. AIAA Journal, vol. 45, no. 9, pages 2125–2139, 2007. (Cited on pages 91 and 93.)

[C.H. Bischof 1998] P.D. Hovland C.H. Bischof A. Carle, P. Khademi and A. Mauer. *ADIFOR 2.0 User's Guide (Revision D)*. Technical Report 192, Argonne National Library, 1998. (Cited on page 83.)

[Clément 1975] Ph. Clément. *Approximation by finite element functions using local regularization*. Revue Française d'Automatique, Informatique et Recherche Opérationnelle, vol. R-2, pages 77–84, 1975. (Cited on page 111.)

[Cournède 2006] P.-H. Cournède, B. Koobus and A. Dervieux. *Positivity statements for a Mixed-Element-Volume scheme on fixed and moving grids*. European Journal of Computational Mechanics, vol. 15, no. 7-8, pages 767–798, 2006. (Cited on page 146.)

[Courty 2005] F. Courty, T. Roy, B. Koobus, M. Vázquez and A. Dervieux. *Error analysis for P1-exact schemes Finite Element for Flow Problems*. 2005. (Cited on pages 112 and 120.)

[D. Guiraud 1988] L. Mane D. Guiraud and P. Leca. *Résolution des Equations de Navier-Stokes 2D et 3D sur des Architectures Multiprocesseurs*. Rapport technique 27–3479, ONERA - Parallélisation de Programmes d'Aérodynamique sur des Installations M.I.M.D., 1988. (Cited on page 189.)

[Debiez 1996] C. Debiez. *Approximation et linéarisation d'écoulements aérodynamiques instationnaires*. PhD thesis, University of Nice, France (in French), 1996. (Cited on pages 23, 26 and 27.)

[Debiez 1998] C. Debiez, A. Dervieux, K. Mer and B. NKonga. *Computation of unsteady flows with mixed finite/finite element upwind methods*. International Journal for Numerical Methods in Fluids, vol. 27, pages 193–206, 1998. (Cited on pages 27 and 29.)

[Debiez 1999] C. Debiez and A. Dervieux. *Mixed Element Volume MUSCL methods with weak viscosity for steady and unsteady flow calculation*. Computer and Fluids, vol. 29, pages 89–118, 1999. (Cited on pages 25, 26 and 27.)

[Desideri 1987] J-A. Desideri, A. Goudjo and V. Selmin. *Third-order numerical schemes for hyperbolic problems*. RR-607, INRIA, 1987. (Cited on page 24.)

[Dompierre 1997] J. Dompierre, M.G. Vallet, M. Fortin, Y. Bourgault and W.G. Habashi. *Anisotropic mesh adaptation: towards a solver and user independent CFD*. AIAA Paper, vol. 97-0861, 1997. (Cited on page 142.)

[Donea 1987] J. Donea, L. Quartapelle and V. Selmin. *An analysis of time discretization in the finite element solution of hyperbolic problems*. J. Comp. Phys., vol. 70, pages 463–499, 1987. (Cited on page 28.)

[Dong 2006] S. Dong, G.E. Karniadakis, A. Ekmekci and D. Rockwell. *A Combined DNS-PIV Study of the Turbulent Near Wake*. Journal of Fluid Mech., vol. 569, pages 187–207, 2006. (Cited on page 65.)

[F. Tremblay 2000] M. Manhart F. Tremblay and R. Friedrich. *DNS of flow around a circular cylinder at a subcritical Reynolds number with cartesian grids.* In Proc. of ETC8, page 659, Barcelona, Spain, 2000. (Cited on page 44.)

[Faure 1998] C. Faure and Y. Papegay. *Odysée Users's Guide version 1.7.* RT-0224, INRIA, 1998. (Cited on page 83.)

[Fidkowski 2011] K.J. Fidkowski and D.L. Darmofal. *Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics: Overview and Recent Results.* AIAA Journal, vol. 49(4), pages 673–694, 2011. (Cited on page 183.)

[Fournier 1998] L. Fournier, G. Carré and S. Lanteri. *Parallel multigrid algorithms for the acceleration of compressible flow calculations.* Calculateurs Parallèles, Réseaux et Systèmes Répartis, vol. 10, no. 4, pages 381–389, 1998. (Cited on page 27.)

[Francescatto 1998] J. Francescatto and A. Dervieux. *A Semi-coarsening Strategy for Unstructured Multigrid Based on Agglomeration.* Int. Journal for Num. Meth. in Fluids, vol. 26, pages 927–957, 1998. (Cited on page 27.)

[Frey 1999] P-J. Frey and P-L. George. Maillages. Hermes, 1999. (Cited on page 130.)

[Frey 2001] P.J. Frey. `Yams`*, A fully automatic adaptive isotropic surface remeshing procedure.* RT-0252, INRIA, nov 2001. (Cited on pages 134 and 170.)

[Frey 2005] P.J. Frey and F. Alauzet. *Anisotropic mesh adaptation for* CFD *computations.* Comput. Methods Appl. Mech. Engrg., vol. 194, no. 48-49, pages 5068–5082, 2005. (Cited on page 142.)

[Frohlich 2008] J. Frohlich and D. von Terzi. *Hybrid LES/RANS methods for the simulation of turbulent flows.* Progress in Aerospace Sciences, vol. 44, pages 349–377, 2008. (Cited on page 54.)

[George 1991] P.L. George, F. Hecht and M.G. Vallet. *Creation of internal points in Vornoi's type method. Control adaptation.* Adv. Eng. Software, vol. 13, no. 5-6, pages 303–312, 1991. (Cited on page 132.)

[Giering 1998] R. Giering and T. Kaminski. *Recipies for adjoint code construction.* ACM Transactions on Mathematical Softwares, vol. 24(4), pages 437–474, 1998. (Cited on page 83.)

[Giles 1997] M.B. Giles. *On adjoint equations for error analysis and optimal grid adaptation in CFD.* Rapport technique NA-97/11, Oxford, 1997. (Cited on page 83.)

[Giles 1999] M.B. Giles and N.A. Pierce. *Improved lift and drag estimates using adjoint Euler equations.* AIAA paper, vol. 99-3293, 1999. (Cited on page 83.)

[Giles 2000] M.B. Giles and N.A. Pierce. *An introduction to the adjoint approach to design.* Flow, Turbulence and Combustion, vol. 65, pages 393–415, 2000. (Cited on page 148.)

[Giles 2002a] M. B. Giles and E. Süli. *Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality.* Acta Numerica,Cambridge University Press, vol. 11, pages 145–236, 2002. (Cited on pages 83 and 98.)

[Giles 2002b] M.B. Giles and E. Suli. Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality, pages 145–236. Cambridge University Press, 2002. (Cited on pages 98 and 143.)

[Gnoffo 1983] P. A. Gnoffo. *A finite-volume, adaptive grid algorithm applied to planetary entry flowfields.* AIAA J., vol. 21:9, pages 1249–1254, 1983. (Cited on page 184.)

[Goldberg 1974] M. Goldberg. *Three Infinite Families of Tetrahedral Space-Fillers.* J. Comb. Theory, Ser. A, vol. 16, no. 3, pages 348–354, 1974. (Cited on page 134.)

[Goldberg 1990] U. Goldberg and D. Ota. *A $k - \varepsilon$ Near-Wall Formulation for Separated Flows.* Rapport technique 91–1482, AIAA 22nd Fluid Dynamics, Plasma Dynamics & Lasers Conference, June 1990. (Cited on page 61.)

[Goldberg 1998] U. Goldberg, O. Peroomian and S. Chakravarthy. *A wall-distance-free $k - \varepsilon$ model with Enhanced Near-Wall Treatment.* Journal of Fluids Engineering, vol. 120, pages 457–462, 1998. (Cited on page 61.)

[Gourvitch 2004] N. Gourvitch, G. Roge, T. Abalakin, A. Dervieux and T. Kozubskaya. *A tetrahedral-based superconvergent scheme for aeroacoustics.* RR-5212, INRIA, 2004. (Cited on page 27.)

[Griewank 1992] A. Griewank. *Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation.* Optimization Methods and Software, vol. 1, pages 35–54, 1992. (Cited on page 89.)

[Gruau 2005] C. Gruau and T. Coupez. *3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric.* Comput. Methods Appl. Mech. Engrg., vol. 194, no. 48-49, pages 4951–4976, 2005. (Cited on page 142.)

[Guégan 2010] D. Guégan, O. Allain, A. Dervieux and F. Alauzet. *An L^∞-L^p mesh adaptive method for computing unsteady bi-fluid flows.* Int. J. Numer. Meth. Eng., 2010. to appear. (Cited on pages 143, 162, 167 and 180.)

[Guerra 1977] F. M. Guerra. *Finite element analysis for the adaptive method for rezoning.* Phd. thesis, The University of Texas Austin, 1977. (Cited on page 4.)

[Guillard 1999] H. Guillard and C. Viozat. *On the behaviour of upwind schemes in the low Mach number limit.* Computers and Fluids, vol. 28, pages 63–86, 1999. (Cited on page 66.)

[H. Baya Toda 2010] K. Truffin H. Baya Toda and F. Nicoud. *Is the dynamic procedure appropriate for all SGS model.* In V European Conference on Computational Fluid Dynamics, ECCOMAS CFD, Lisbon, Portugal, June 2010. (Cited on page 54.)

[Habashi 2000] W.G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin and M.-G. Vallet. *Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent cfd solutions: Part I: General principles.* International Journal for Numerical Methods in Fluids, vol. 32, pages 725–744, 2000. (Cited on page 130.)

[Hascoët 2005] L. Hascoët and V. Pascual. *Extension of TAPENADE towards Fortran 95.* Automatic Differentiation: Applications, Theory, and Tools, Lecture Notes in Computational Science and Engineering, 2005. (Cited on page 83.)

[Hascoët 2008] L. Hascoët and V. Pascual. *TAPENADE for C.* Advances in Automatic Differentiation, Springer, 2008. (Cited on page 83.)

[Hecht 1997] F. Hecht and B. Mohammadi. *Mesh adaptation by metric control for multi-scale phenomena and turbulence.* In 35th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-1997-0859, Reno, NV, USA, Jan 1997. (Cited on page 142.)

[Hinze 1959] J. Hinze. Turbulence. McGraw-Hill, 1959. (Cited on page 60.)

[Hirsch 1988] C. Hirsch. Numerical computation of internal and external flows. J. Wiley & Sons, Chichester., 1988. (Cited on page 23.)

[Huang 2010] W. Huang and X. Li. *An anisotropic mesh adaptation method for the finite element solution of variational problems.* Finite Elements in Analysis and Design, vol. 46, pages 61–73, 2010. (Cited on page 110.)

[Jameson 1987] A. Jameson. *Successes and challenges in computational aerodynamics.* AIAA paper 1987-1184, 1987. AIAA 8th Computational Fluid Dynamics Conference, Honolulu, Hawaii. (Cited on page 82.)

[Jameson 1993] A. Jameson. *Artificial Diffusion, Upwind Biasing, Limiters and their Effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flows.* AIAA journal, vol. 93-3359, 1993. (Cited on page 23.)

[John 2006] V. John. *On Large Eddy Simulation and Variational Multiscale methods in the numerical simulation of turbulent incompressible flows.* Applications of Mathematics, vol. 51, pages 321–353, 2006. (Cited on page 50.)

[Johnson 1995] C. Johnson. *On computatbility and error control in CFD.* International Journal for Numerical Methods in Fluids, vol. 20:8-9, page 777–788, 1995. (Cited on page 184.)

[Jones 1969] G.W. Jones, J. Cincotta and R.W. Walker. *Aerodynamic Forces on a stationnary and oscillating circular cylinder at high Reynolds numbers.* Research Report R-300, NASA, 1969. (Cited on page 71.)

[Knoll 2004] D. A. Knoll and D. E. Keyes. *Jacobian-free Newton-Krylov methods: a survey of approaches and applications.* J. Comp. Phys., vol. 193, no. 2, pages 357–397, 2004. (Cited on page 27.)

[Koobus 2004] B. Koobus and C. Farhat. *A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes-application to vortex shedding.* Comput. Methods Appl. Mech. Eng, vol. 193, pages 1367–1383, 2004. (Cited on pages 50 and 51.)

[Koobus 2007] B. Koobus, S. Camarri, M.V. Salvetti, S. Wornom and A. Dervieux. *Parallel simulation of three-dimensional flows: application to turbulent wakes and two-phase compressible flows.* Advances in Engineering Software, vol. 38, pages 328–337, 2007. (Cited on page 57.)

[Kravchenko 2000] A. Kravchenko and P. Moin. *Numerical studies of flow over a circular cylinder at* $Re_D = 3900$. Phys. of Fluids, vol. 12:2, pages 403–417, 2000. (Cited on page 65.)

[Krizek 1998] M. Krizek and P. Neittaanmaki. Bibliography on superconvergence, volume 196 of *Lecture Notes in Pure and Applied Mathematics*, pages 315–348. Springer-Verlag, 1998. (Cited on pages 98 and 113.)

[Lallemand 1992] M. H. Lallemand, H. Steve and A. Dervieux. *Unstructured multigridding by volume agglomeration : current status.* Comput. Fluids, vol. 21, pages 397–433, 1992. (Cited on pages 27 and 51.)

[Launder 1979] B.E. Launder and D.B. Spalding. *The numerical computation of turbulent flows.* Comp. Meth. Appl. Mech. and Eng., vol. 3, pages 269–289, 1979. (Cited on pages 57 and 60.)

[Leer 1977] B. Van Leer. *Towards the Ultimate Conservative Difference Scheme IV: A new approach to numerical convection.* J. Comp. Phys., vol. 23, pages 276–299, 1977. (Cited on pages 18 and 21.)

[Leer 1979] B. Van Leer. *Towards the Ultimate Conservative Difference Scheme V: A Second Order Sequel to Godunov's Method.* J. Comp. Phys., vol. 32, pages 101–136, 1979. (Cited on page 18.)

[Li 2005] X. Li, M.S. Shephard and M.W. Beal. *3D anisotropic mesh adaptation by mesh modification.* Comput. Methods Appl. Mech. Engrg., vol. 194, no. 48-49, pages 4915–4950, 2005. (Cited on page 142.)

[Lilly 1992] D. K. Lilly. *A proposed modification of the Germano subgrid scale closure model*. Physics of Fluids, vol. 4, no. 633, 1992. (Cited on pages 52 and 53.)

[Lions 1968] J.L. Lions. Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles. Dunod, Paris, 1968. English trans. Springer-Verlag Berlin 1971. (Cited on page 81.)

[Löhner 1989] R. Löhner. *Adaptive Remeshing for Transient Problems*. Comput. Methods Appl. Mech. Engrg., vol. 75, pages 195–214, 1989. (Cited on page 143.)

[Loseille 2007] A. Loseille, A. Dervieux, P.J. Frey and F. Alauzet. *Achievement of global second-order mesh convergence for discontinuous flows with adapted unstructured meshes*. AIAA paper, vol. 2007-4186, 2007. (Cited on pages 118, 142, 143, 163 and 164.)

[Loseille 2008] A. Loseille. *Adaptation de maillage anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides. Application à la prédiction haute-fidélité du bang sonique*. PhD thesis, Université Pierre et Marie Curie, Paris VI, Paris, France, 2008. (in French). (Cited on pages 4, 118 and 137.)

[Loseille 2009] A. Loseille and F. Alauzet. *Continuous mesh model and well-posed continuous interpolation error estimation*. RR-6846, INRIA, march 2009. (Cited on pages 121 and 142.)

[Loseille 2010a] A. Loseille, A. Dervieux and F. Alauzet. *Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations*. J. Comput. Phys., 2010. Accepted. doi:10.1016/j.jcp.2009.12.021. (Cited on pages 122, 143, 145, 146, 150, 154, 155, 185 and 187.)

[Loseille 2010b] A. Loseille and R. Löhner. *Adaptive anisotropic simulations in aerodynamics*. In 48th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2010-169, Orlando, FL, USA, Jan 2010. (Cited on page 170.)

[Loseille 2011a] A. Loseille and F. Alauzet. *Continuous mesh framework. Part I: well-posed continuous interpolation error*. SIAM, vol. 49, no. 1, pages 38–60, 2011. (Cited on pages 152 and 153.)

[Loseille 2011b] A. Loseille and F. Alauzet. *Continuous mesh framework. Part II: validations and applications*. SIAM, vol. 49, no. 1, pages 61–86, 2011. (Cited on pages 152 and 153.)

[L.S. Pontryagin 1962] R.V. Gamkrelidze L.S. Pontryagin V.G. Boltyanskii and E.F. Mischenko. The mathematical theory of optimal processes, volume 4. Interscience, 1962. Translation from Russian Book. (Cited on page 81.)

[M. Germano 1991] P. Moin M. Germano U. Piomelli and W.H. Cabot. *A Dynamic Subgrid-Scale Eddy Viscosity Model.* Physics of Fluids, vol. 3, pages 1760–1765, 1991. (Cited on pages 53 and 54.)

[Martin 1996] R. Martin and H. Guillard. *A second-order defect correction scheme for unsteady problems.* Comput. and Fluids, vol. 25, no. 1, pages 9–27, 1996. (Cited on pages 23 and 27.)

[Martinelli 2010a] M. Martinelli, A. Dervieux, L. Hascoët, V. Pascual and A. Belme. *AD-based perturbation methods for uncertainties and errors.* Int. J. Engineering Systems Modelling and Simulation, vol. 2, no. 1/2, pages 65–74, 2010. (Cited on pages iv, 83, 95, 96, 102, 103 and 104.)

[Martinelli 2010b] M. Martinelli and R. Duvigneau. *On the use of second-order derivative and metamodel-based Monte-Carlo for uncertainty estimation in aerodynamics.* Computers and Fluids, vol. 37, no. 6, 2010. (Cited on page 95.)

[Mavriplis 1990] D. Mavriplis. *Turbulent Flow Calculations Using Unstructured and Adaptive Meshes.* International Journal for Numerical Methods in Fluids, vol. 13:9, page 1131–1152, 1990. (Cited on page 184.)

[Mavriplis 1997] D.J. Mavriplis. *Unstructured Grid Techniques.* Annual Review of Fluid Mechanics, vol. 29, pages 473–514, 1997. (Cited on page 27.)

[Menter 2004] F.R. Menter and M. Kuntz. *Adaptation of eddy-viscosity turbulence models to unsteady separated flow behind vehicles.* Symposium on the aerodynamics of heavy vehicles: trucks, buses and trains, Springer Berlin Heidelberg New York, 2004. (Cited on page 64.)

[Mer 1998] K. Mer. *Variational analysis of a mixed element volume scheme with fourth-order viscosity on general triangulations.* Comput. Methods Appl. Engrg., vol. 153, pages 45–62, 1998. (Cited on page 16.)

[N. Gauger 2007] C. Moldenhauer N. Gauger A. Walther and M. Widhalm. *Automatic Differentiation of an entire design chain with applications.* New Advances in Num. and Exp. Fluid Mechanics, 2007. (Cited on page 83.)

[Nakao 1998] M. T. Nakao and N. Yamamoto. *Constructive L^2 Error Estimates for Finite Element Solutions of the Stokes Equations.* Reliable Computing, vol. 4, pages 115–124, 1998. (Cited on page 110.)

[Nicoud 1999] F. Nicoud and F. Ducros. *Subgrid-scale stress modelling based on the square of the velocity gradient tensor.* Flow Turb. Comb, vol. 62, no. 3, pages 183–200, 1999. (Cited on page 52.)

[Nielsen 2004] E. Nielsen, J. Lu, M. Park and D. Darmofal. *An Implicit, Exact Dual Adjoint Solution Method for Turbulent Flows on Unstructured Grids.* Computers and Fluids, vol. 33:9, pages 1131–1155, 2004. (Cited on page 184.)

[Norberg 1987] C. Norberg. *Effect of Reynolds number and a low-intensity freestream turbulence on the flow around a circular cylinder*. Dept. Applied Thermodynamics and Fluid Mechanics, Chalmers University of Technology, 1987. (Cited on page 65.)

[Norberg 2003] C. Norberg. *Fluctuating lift on a circular cylinder: review and new measurements*. J. Fluids Struct, vol. 17, pages 57–96, 2003. (Cited on pages 67 and 68.)

[Olivier 2011] G. Olivier. *Anisotropic metric-based mesh adaptation for unsteady CFD simulation involving moving geometries*. PhD thesis, Université Pierre et Marie Curie, Paris VI, Paris, France, 2011. (Cited on page 195.)

[Ong 1996] L. Ong and J. Wallace. *The velocity field of the turbulent very near wake of a circular cylinder*. Exp. in Fluids, vol. 20, pages 441–453, 1996. (Cited on pages 65 and 67.)

[Ouvrard 2009] H. Ouvrard. *Simulation numérique d'écoulements turbulents pas ler approches LES, VMS-LES et Hybride RANS/VMS-LES en maillages non-structurés*. PhD thesis, University of Montpellier II, France (in French), 2009. (Cited on page 69.)

[Ouvrard 2010] H. Ouvrard, B. Koobus, A. Dervieux and M. V. Salvetti. *Classical and variational multiscale* LES *of the flow around a circular cylinder on unstructured grids*. Comp. Fluids, pages 1083–94, 2010. (Cited on pages 50, 65 and 67.)

[P. Batten 2000] U. Goldberg P. Batten and S. Chakravarthy. *SubGrid Turbulence Modeling for Unsteady Flow with Acoustic Resonance*. In 38th Aerospace Sciences Meeting - AIAA, Reno, NV, USA, 2000. (Cited on pages 45 and 57.)

[P. Batten 2004] U. Goldberg P. Batten and S. Chakravarthy. *Interfacing statistical turbulence closures with large-eddy simulation*. AIAA Journal, vol. 42, pages 485–492, 2004. (Cited on page 57.)

[P. Hovland 1998] B. Mohammadi P. Hovland and C. Bischof. *Automatic Differentiation of Navier-Stokes computations*. Comput. Methods for Optimal Design and Control, pages 265–284, 1998. (Cited on page 83.)

[P. Parneaudeau 2008] D. Heitz P. Parneaudeau J. Carlier and E. Lamballais. *Experimental and numerical studies of the flow over a circular cylinder at Reynolds number* 3900. Phys. of Fluids, vol. 20, 2008. (Cited on pages 65, 67, 69 and 70.)

[Pain 2001] C.C Pain, A.P. Humpleby, C.R.E. de Oliveira and A.J.H. Goddard. *Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations*. Comput. Methods Appl. Mech. Engrg., vol. 190, pages 3771–3796, 2001. (Cited on page 142.)

[Palmerio 1996] B. Palmerio. *Coupling mesh and flow in viscous fluid calculations on unstructured triangular finite element.* Computuational Fluid Dynamics, vol. 6, pages 275–290, 1996. (Cited on page 130.)

[Park 2003] M. A. Park. *Three-Dimensional Turbulent RANS Adjoint-Based Error Correction.* In AIAA-2003-3849, June 2003. (Cited on page 184.)

[Park 2010] M. A. Park and J. R. Carlson. *Turbulent Output-Based Anisotropic Adaptation.* In AIAA-2010-0168, January 2010. (Cited on page 184.)

[Peraire 1987] J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz. *Adaptive Remeshing for Compressible Flow Computations.* J. Comput. Phys., vol. 72, pages 449–466, 1987. (Cited on page 142.)

[Perotto 2008] S. Perotto and S. Micheletti. *Output functional control for nonlinear equations driven by anisotropic mesh adaptation.* SIAM J. Sci. Comput., vol. 30, pages 2817–2854, 2008. (Cited on page 110.)

[Pierce 2000] N.A. Pierce and M.B. Giles. *Adjoint Recovery of Superconvergent Functionals from PDE Approximations.* SIAM Review, vol. 42, no. 2, pages 247–264, 2000. (Cited on pages 83 and 98.)

[Pierce 2004] N. A. Pierce and M. B. Giles. *Adjoint and defect error bounding and correction for functional estimates.* Journal of Computational Physics, vol. 200, pages 769–794, 2004. (Cited on pages 83, 95, 98 and 100.)

[Pironneau 1974] O. Pironneau. *On optimum design in fluid mechanics.* J. Fluid Mech., vol. 64, pages 97–110, 1974. (Cited on page 82.)

[P.R. Spalart 1997] M. Strelets P.R. Spalart W-H. Jou and S.R. Allmaras. *Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach.* Advances in DNS/LES, pages 137–147, 1997. (Cited on pages 45, 54 and 55.)

[P.R. Spalart 2006] M. Strelets P.R. Spalart S. Deck, M.L. Shur, A. Travin and K.D. Squires. *A new version of detached-eddy simulation, resistant to ambiguous grid densities.* Theor. Comput. Fluid Dyn., vol. 20, pages 181–195, 2006. (Cited on page 56.)

[Pulliam 1989] T.H. Pulliam. *Low Reynolds number numerical solution of Chaotic Flows.* AIAA Journal, vol. 89, no. 0123, 1989. (Cited on page 189.)

[Roshko 1961] A. Roshko. *Experiments on the Flow past a Circular Cylinder at very high Reynolds numbers.* J. Fluid Mech., vol. 10-3, pages 345–356, 1961. (Cited on pages 70 and 71.)

[S. Ghosal 1995] P. Moin S. Ghosal T. Lund and K. Akselvoll. *The dynamic localization model for large eddy simulation of turbulent flows.* J. Fluid Mech., vol. 286, pages 229–255, 1995. (Cited on page 53.)

[Sagaut 2002] P. Sagaut and E. Labourasse. *Reconstruction of turbulent fluctuations using a hybrid RANS/LES approach.* J. Comp. Phys., vol. 182, pages 301–336, 2002. (Cited on pages 54, 58 and 59.)

[Sarkis 2000] M. Sarkis and B. Koobus. *A scaled and minimum overlap restricted additive Schwarz method with application on aerodynamics.* Comput. Methods Appl. Mech. Eng., vol. 184, pages 391–400, 2000. (Cited on page 27.)

[Schewe 1983] G. Schewe. *On the force fluctuations acting on a circular cylinder in crossflow from subcritical up to transcritical Reynolds numbers.* J. of Fluid Mechanics, vol. 133, pages 265–285, 1983. (Cited on pages 71, 74 and 77.)

[Selmin 1998] V. Selmin and L. Formaggia. *Unified construction of finite element and finite volume discretizations for compressible flows.* International Journal for Numerical Methods in Engineering, vol. 39, no. 1, pages 1–32, 1998. (Cited on page 13.)

[Shih 1993] W.C. L. Shih, C. Wang, D. Coles and A. Roshko. *Experiments on Flow past rough circular cylinders at large Reynolds numbers.* J. Wind Engg. and Industrial Aerodynamics, vol. 49, pages 351–368, 1993. (Cited on pages 71 and 77.)

[Smagorinsky 1963] J. Smagorinsky. *General circulation experiments with the primitive equations.* Month. Weath. Rev, vol. 91, no. 3, pages 99–164, 1963. (Cited on page 52.)

[Sommerville 1923] D.Y.M. Sommerville. *Space-filling tetrahedra in Euclidean space.* In Proc. Edinburgh Math. Soc., volume 41, pages 49–57, 1923. (Cited on page 134.)

[Steger 1981] J. L. Steger and R. F. Warming. *Flux Vector Splitting of the Inviscid Gasdynamic Equations with Applications to Finite-Difference Methods.* J. of Comp. Physics, vol. 40, pages 263–293, 1981. (Cited on pages 41 and 66.)

[Stoufflet 1996] B. Stoufflet, J. Periaux, L. Fezoui and A. Dervieux. *3-D Hypersonic Euler Numerical Simulation around Space Vehicles using Adapted Finite Elements.* 25th AIAA Aerospace Meeting, Reno (1987), AIAA paper 86-0560, 1996. (Cited on page 13.)

[T. Strouboulis 2000] D.K. Datta T. Strouboulis I. Babuska, K. Copps and S.K. Gangaraj. *A posteriori estimation and adaptive control of the error in the quantity of interest. Part I : A posteriori estimation of the von Mises stress and the stress intensity factor.* Comput. Meth. Appl. Mech. Engrg., vol. 181, pages 261–294, 2000. (Cited on page 4.)

[Tam 2000] A. Tam, D. Ait-Ali-Yahia, M.P. Robichaud, M. Moore, V. Kozel and W.G. Habashi. *Anisotropic mesh adaptation for 3D flows on structured and* *unstructured grids.* Comput. Methods Appl. Mech. Engrg., vol. 189, pages 1205–1230, 2000. (Cited on page 142.)

[T.J.R. Hughes 2000] L.Mazzei T.J.R. Hughes and K.E. Jansen. *Large-eddy simulation and the variational multiscale method.* Comput. Vis. Sci., vol. 3, pages 47–59, 2000. (Cited on pages 44 and 62.)

[Travin 1999] A. Travin, M. Shur, M. Strelets and P. Spalart. *Detached-Eddy Simulations Past a Circular Cylinder.* Flow, Turbulence and Combustion, vol. 63, pages 293–313, 1999. (Cited on pages 4, 54, 71 and 78.)

[van Nunen 1974] J.W.G. van Nunen. *Pressure and forces on a circular cylinder in a cross flow at high Reynolds numbers.* In:Naudascher, (Ed.) Flow Induced Structural Vibrations, pages 748–754, 1974. (Cited on pages 71 and 78.)

[Venditti 2002] D.A. Venditti and D.L. Darmofal. *Grid adaptation for functional outputs: application to two-dimensional inviscid flows.* J. Comput. Phys., vol. 176, no. 1, pages 40–69, 2002. (Cited on page 110.)

[Venditti 2003a] D.A. Venditti and D.L. Darmofal. *Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows.* J. Comput. Phys., vol. 187, no. 1, pages 22–46, 2003. (Cited on page 110.)

[Venditti 2003b] D.A. Venditti and D.L. Darmofal. *Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows.* Journal of Computational Physics, vol. 187, pages 22–46, 2003. (Cited on page 184.)

[Vengadesan 2007] S. Vengadesan and P. Nithiarasu. *Hybrid LES - Review and assessment.* Sadhana, vol. 32, pages 501–511, 2007. (Cited on page 54.)

[Verfürth 1996] R. Verfürth. A review of *a posteriori* error estimation and adaptative mesh-refinement techniques. Wiley Teubner Mathematics, New York, 1996. (Cited on page 143.)

[Wang 2001] M. Wang, P. Catalano and G. Iaccarino. *Prediction of high Reynolds number the flow over a circular cylinder using LES.* Annual research briefs, Center for Turbulence Research, Stanford, 2001. (Cited on pages 71, 77 and 78.)

[Warschauer 1971] K.A. Warschauer and J.A. Leene. *Experiments on mean and fluctuating pressures of circular cylinders at cross-flow at very high Reynolds numbers.* In Proc. Int. Conf. on Wind Effects on Buildings and Structures, pages 305–315, Saiko, Tokyo, 1971. (Cited on page 71.)

[Wintzer 2008] M. Wintzer, M. Nemec and M.J. Aftosmis. *Adjoint-Based Adaptive Mesh Refinement for Sonic Boom Prediction.* In AIAA 26th Applied Aerodynamics Conference, AIAA-2008-6593, Honolulu, HI, USA, Aug 2008. (Cited on page 143.)

[Wu 1995] H. Wu and L. Wang. *Non-existence of third order accurate semi-discrete MUSCL-type schemes for nonlinear conservation laws and unified construction of high accurate ENO schemes.* Sixth International Symposium on Computational Fluid Dynamics, Lake Tahoe, NV, 1995. (Cited on page 20.)

[Yakhot 1986] V. Yakhot and S. A. Orszag. *Renormalization group analysis of turbulence. I. Basic theory.* Journal of Scientific Computing, vol. 1, no. 1, pages 3–51, 1986. (Cited on page 60.)

[Zdravkovich 1997] M. Zdravkovich. Flow around circular cylinders vol 1: Fundamentals. Oxford University Press, 1997. (Cited on pages 71 and 77.)

[Zienkiewicz 1992a] O. C. Zienkiewicz and J. Z. Zhu. *The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique.* Int. J. Numer. Meth. Engng, vol. 33, no. 7, pages 1331–1364, 1992. (Cited on pages 98 and 113.)

[Zienkiewicz 1992b] O. C. Zienkiewicz and J. Z. Zhu. *The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity.* Int. J. Numer. Meth. Engng, vol. 33, no. 7, pages 1365–1380, 1992. (Cited on page 113.)

Aérodynamique instationnaire et méthode adjointe

Resumé: Cette thèse contribue à la simulation numérique des écoulements d'un fluide compressible modélisé par les équations de Euler et Navier-Stokes: étude d'un schéma d'ordre élévé basé sur une matrice de masse, modélisation des écoulement turbulents compressibles à très haut Reynolds, développement des estimateurs et correcteurs d'erreurs *a posteriori* et *a priori*, et adaptation de maillage anisotrope pour les fonctionnelles d'observation. Concernant la prédiction des écoulements turbulents, on s'est intéressé aux modèles hybrides de type RANS/LES comportant les nouveautés suivantes: traitement des tourbillons de grande échelle utilisant la formulation VMS (Variational Multi-Scale) et du RANS employé sur la paroi sur une distance imposée via une zone de protection conçue afin d'éviter le phénomène assez commun apelé "grid induced model depletion". Le niveau de viscosité du modèle VMS-LES est de plus controllé par un procédé de double filtre dynamique. La seconde partie concerne l'adaptation de maillage anisotrope pour mieux observer une fontionnelle d'observation. Les estimations *a priori* sont réalises pour le modèle des équations d'Euler et Navier-Stokes en instationnaire en 2D et 3D. A partir de ces estimations on sait définir les maillages optimaux au cours du calcul instationnaire, en fonction de l'état et de l'état adjoint. Le système d'optimalité est discrétisé et résolu à l'aide d'une méthode de point fixe instationnaire global, comportant une stratégie de stockage/recalcul pour le couplage état/ état adjoint. Des applications à la propagation d'ondes de choc et d'ondes acoustiques sont présentées.

Mots-clés: écoulements compressibles, Navier-Stokes, modèles de turbulence, estimateur d'erreur, adjoint instationnaire, adaptation de maillage, anisotropie, fonctionelle

Unsteady Aerodynamics and Adjoint Method

Abstract: In this thesis, we first focused on error estimates for unsteady problems. We have contributed to both a posteriori and a priori error estimators for unsteady inviscid problems and viscous unsteady problems. For the first one, we have been interested on linearized methods for reducing dissipation errors. Regarding the a priori errors, a new estimator is proposed with application to viscous compressible flows. These a priori estimators have been employed for goal-oriented anisotropic mesh adaptation problems, for both Euler and laminar Navier-Stokes flows, in a joint work with Gamma3 team We have developped a method to derive an optimal mesh to observe/improve a given output functional in an unsteady context. The weights of the interpolation error are adjoint states in this case. A new global fixed-point algorithm is proposed herein order to converge the couple mesh/solution. We have applied this algorithm for blast wave problems and acoustics, for both 2D and 3D cases.

Keywords: compressible flow, Navier-Stokes, turbulence models, error estimation, unsteady adjoint, anisotropic mesh adaptation, functional