Numerical approximation and adaptation for flows in rotating machines

Bastien Sauvage

To cite this version:

Bastien Sauvage. Numerical approximation and adaptation for flows in rotating machines. Numerical Analysis [math.NA]. Université Côte d'Azur, 2024. English. NNT : 2024COAZ5045 . tel-04810649

HAL Id: tel-04810649

https://theses.hal.science/tel-04810649v1

Submitted on 29 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ
CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES FONDAMENTALE
ET APPLIQUÉES

THÈSE DE DOCTORAT

Approximation et adaptation numériques pour les écoulements en machines tournantes

Bastien SAUVAGE

Centre Inria d'Université Côte d'Azur,
ECUADOR

Présentée en vue de l'obtention
du grade de docteur en Mathématiques
d'Université Côte d'Azur

Dirigée par : Boniface NKONGA

Co-Direction : Alain DERVIEUX

Soutenue le : 27 Septembre 2024

Devant le jury, composé de :

Anca BELME, Maître de conférences, Sorbonne Université

Marianna BRAZA, Directeur de recherche, IMFT

Frédéric ALAUZET, Directeur de recherche, INRIA Saclay

Thierry COUPEZ, Professeur, MinesParis PSL

Bruno KOOBUS, Professeur, Université de Montpellier

anr
agence nationale
de la recherche

Approximation et adaptation numériques pour les écoulements en machines tournantes

Jury :

Rapporteurs

Anca BELME,	Maître de conférences,	Sorbonne Université
Marianna BRAZA,	Directeur de recherche,	IMFT

Examinateurs

Frédéric ALAUZET,	Directeur de recherche,	INRIA Saclay
Thierry COUPEZ,	Professeur,	MinesParis PSL
Bruno KOOBUS,	Professeur,	Université de Montpellier

Approximation et adaptation numériques pour les écoulements en machines tournantes

---

Ce travail s'inscrit dans un projet de recherche dont l'objectif est de proposer des simulations numériques en fluides (CFD) capables de capturer le bruit produit par les machines tournantes.

Cette thèse, au sein de ce projet, à pour objectif d'étudier des méthodes d'adaptation de maillages (instationnaires) pour des écoulements turbulents instationnaires. On commence par décrire les outils necessaires à nos études, à savoir les méthodes numériques, les modèles de turbulences et l'adaptation de maillages. Une première partie est consacrée à l'étude de l'algorithme d'adaptation de maillage instationnaire "Transient Fixed Point" appliqué à des écoulements turbulents autour de cylindres à différents nombre de Reynolds. Puis on étudie une méthode pour prendre en compte la rotation dans les simulations, que l'on couple avec les méthodes d'adaptation de maillage. Des exemples numériques sont proposés. Ces premiers travaux se heurtent à deux importants problèmes ouverts en CFD maillage-adaptative.

Afin de choisir de manière optimale le pas d'avancement en temps implicite, on présente une nouvelle méthode d'adaptation de maillages espace-temps, qui permet d'adapter simultanément le maillage en espace et le maillage en temps. Afin d'adapter le maillage à la fois à l'ecoulement moyen et aux grandes structures turbulentes, on propose une nouvelle approche d'adaptation de maillage pour la turbulence en calcul LES et hybride.

Mots clés : calcul scientifique, mécanique des fluides numérique, adaptation de maillages, turbulence, simulation des grandes structures, écoulements instationnaires, simulations rotor/stator

---

Numerical approximation and adaptation for flows in rotating machines

---

This work is part of a research project aimed at proposing numerical fluid simulations (CFD) capable of capturing the noise produced by rotating machines.

The aim of this thesis is to study mesh adaptation methods for unsteady turbulent flows. We begin by describing the tools required for our studies, namely numerical methods, turbulence models and mesh adaptation. The first part is devoted to the study of the "Transient Fixed Point" unsteady mesh adaptation algorithms applied to turbulent flows around cylinders at different Reynolds numbers. A method for taking rotation into account in simulations is then studied, and coupled with mesh the adaptation methods. Numerical examples are proposed.

This initial work encounted two major unsolved problems in CFD mesh-adaptation. In order to optimally select the implicit time step, a new space-time mesh adaptation method is presented, which simultaneously adapts the space mesh and the time mesh. In order to adapt the mesh to both mean flow and large turbulent structures, a new mesh adaptation approach is proposed for turbulence in LES and hybrid computation.

Keywords : scientific computing, Computational Fluid Dynamics, mesh adaptation, turbulence, Large Eddy Simulation, unsteady flows, rotor/stator simulations

Table des matières

CHAPITRE 1

Introduction

Numerical simulation is an essential tool for predicting and understanding the physical phenomena that surround us. These simulations are based on discretized versions of partial differential equations (PDEs), which are the result of mathematical modeling and must provide the best possible description of the phenomena observed.

These mathematical models are the subject of extensive theoretical work to study their properties. In particular, we want to show that our PDE or system of PDEs has a solution, that this solution is unique and that it depends continuously on the initial conditions. If this is the case, we say that our problem is well-posed in Hadamard's sense. These results provide valuable information on the mathematical and physical properties of the solutions, but it should be pointed out that this theoretical work is long and very difficult. The unfortunate result is that the majority of complex PDE models do not have such results. These questions will not be addressed in this thesis.

There is thus a more applied branch of the study of PDEs, the numerical analysis of partial differential equations (convergence, stability, consistency and accuracy).

This thesis is part of a more applied field known as CFD (Computational Fluid Dynamics), the science of simulating numerically fluid flows. In this field, we restrict ourselves mainly to the discrete forms of the Euler and Navier-Stokes equations, compressible or incompressible, depending on the type of flow studied.

Our CFD application context is to be found in the Norma project, in which this thesis is included. The context of this project lies in the fact that the ecology of urban/extra-urban areas increasingly deteriorated by noise emission generated by rotating machines (e.g. helicopters). The Norma project team will therefore be investigating numerical methods for producing simulations of rotating machines whose noise can be captured numerically.

In particular, this thesis will contribute to numerical methods for taking rotation into account in numerical simulations and to mesh adaptation methods for turbulent flows. More precisely, as aerodynamic noise is produced by pressure fluctuations, which themselves come from the turbulence generated around bodies. We will focus on the study of mesh adaptation methods for unsteady turbulent flows. Let us see how these works are organized in the thesis.

Organization

We start, in Chapter 2, with a reminder of the numerical methods that will be used in the rest of the thesis. We will present a mixed finite-Elements/finite-Volume formulation to discretize the compressible Navier-Stokes equations in space, as well as an extension of our scheme to second order on an unstructured mesh.

Turbulence simulation occupies an important place in this thesis, so Chapter 3 will review the main turbulence models we will be using for our simulations. In particular, for RANS models, we will introduce the Spalart-Allmaras model, which will be the only RANS model used in this thesis. Then we will set out the various LES, VMS-LES formulations that will be of major use in our studies (mesh adaptation for unsteady turbulent flows).

Chapter 4, the last chapter in the series of reminders, deals with the notion of mesh adaptation, which will be central to this thesis. The notion of mesh adaptation chosen is based on the notion of metrics and continuous mesh, the development of which can be found in [82] and [83].

Chapter 5 presents a method for polynomial reconstruction of the numerical solution, called Central Essentially Oscillating (CENO) approximation, which allows us to recover a solution of order 3 or 4.

Our first study, in Chapter 6, concerns a particular use of the Transient Fixed Point mesh adaptation algorithm applied to flows around cylinders at different Reynolds numbers.

Then, in Chapter 7, we present several methods for taking rotation into account in our numerical simulations, namely the Chimera method, Single Reference Frame and Multiple Reference Frame. The chosen method, Multiple Reference Frame, is then applied to the Caradonna-Tung [29] test case with mesh adaptation.

A new theory of space-time mesh adaptation is given Chapter 8. This new adaptation method enables us to find the optimal implicit time steps for our unsteady flows, coupled with the spatially optimal mesh. This algorithm is based on balancing the two errors, those in space and those in time, which gives us a good distribution of points in space and time for a given space-time complexity. This algorithm will be tested, in 2D and 3D, on URANS flows around cylinders at Reynolds 3900 and 1M.

Finally, in Chapter 9, we introduce a new error analysis for mesh adaptation for LES flows. This work is based on the work of S. Toosi and J. Larsson [130], [131], and allows us to control, in addition to the error on a chosen sensor, the error committed by LES modeling by controlling the (optimal) filter size thanks to a new optimal metric.

Contributions

During this thesis I worked on and implemented in two different codes : NiceFlow® (LEMMA company, C++ language) and Wolf (GAMMAO Team INRIA, C language). In the first, I helped implement the Multiple Reference Frame method and the new LES adaptation criterion. In Wolf I implemented our new space-time mesh adaptation algorithm, which is an extension of Global Transient Fixed Point.

scientific communications

Journal paper

B. Sauvage, F. Alauzet, A. Dervieux, *A space and time fixed point mesh adaptation method*, Journal of Computational Physics, Volume 519, 113389, 2024.

Preprints

B. Sauvage, F. Miralles, S. Wornom, B. Koobus, A. Dervieux, *Computing the flow past a cylinder : influence of models and numerics.*

B. Sauvage, B. Koobus, F. Alauzet, A. Dervieux, *A metric-based mesh adaptation for hybrid RANS/LES flow calculations.*

Research report

B. Sauvage, D. Chargy, *A mesh adaptative method for rotating machines*, Inria Research Report n°9471, June 2022.

F. Miralles, B. Sauvage, S. Wornom, B. Koobus, A. Dervieux, *Hybrid RANS/DVMS modeling for static and rotating obstacles*, Inria Research Report n°9464, March 2022.

Conferences

F. Miralles, B. Sauvage, A. Duben, V. Bobkov, T. Kozubskaya, S. Wornom, B. Koobus, A. Dervieux, *Simulation of massively separated flows and rotating machine flows using hybrid models*, The 8th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS Congress 2022, 5-9 June, Oslo, Norway.

F. Miralles, B. Sauvage, S. Wornom, B. Koobus, A. Dervieux, *Application of hybrid RANS/VMS modeling to rotating machines*, Conference on Modelling Fluid Flow (CMFF'22), August 30 - September 2, 2022, Budapest, Hungary.

F. Miralles, B. Sauvage, S. Wornom, F. Alauzet, B. Koobus, A. Dervieux, *Simulation of massively separated flows using hybrid models and mesh adaptation*, Direct and Large-Eddy Simulation 13, October 26th-29th, Udine, Italy.

F. Miralles, B. Sauvage, S. Wornom, B. Koobus, A. Dervieux, *Assessment of turbulence hybrid models with transition modeling for the simulation of massively separated flows*, 22nd IACM Computational Fluids Conference, CFC 2023, 25-28 April, Cannes, France.

B. Sauvage, F. Miralles, S. Wornom, B. Koobus, F. Alauzet, A. Dervieux, *Assessment of mesh adaptation algorithms for LES and DES simulation of detached flows*, 22nd IACM Computational Fluids Conference, CFC 2023, 25-28 April, Cannes, France.

L. Abalakin, V. Bobkov, V. Tsvetkova, B. Sauvage, F. Miralles, T. Kozubskaya, S. Wornom, B. Koobus, A. Dervieux, *Towards efficient simulation of turbulent flows and noise in rotating machines*, 22nd IACM Computational Fluids Conference, CFC 2023, 25-28 April, Cannes, France.

B. Sauvage, F. Miralles, S. Wornom, B. Koobus, F. Alauzet, A. Dervieux, *About mesh adaptation for hybrid flow calculation*, International ERCOFTAC Symposium on Engineering Turbulence Modelling and Measurements, Barcelona, september 6-8 2023, Spain.

CHAPITRE 2

Numerical method

1 Introduction

In the context of this thesis, we will only focus on compressible flows, so let us start by recalling the equations used to describe the motion of compressible Newtonian fluids, namely the compressible Navier-Stokes equations in the particular case of a calorically perfect gas :

$$\begin{cases} \partial_t \rho + \operatorname{div}(\rho \mathbf{u}) = 0, \\ \partial_t(\rho \mathbf{u}) + \operatorname{div}(\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = \operatorname{div} \mathcal{T}, \\ \partial_t(\rho E) + \operatorname{div}((\rho E + p)\mathbf{u}) = \operatorname{div}(\lambda \nabla T + \mathcal{T} \cdot \mathbf{u}), \end{cases}$$

where ρ denotes the density (kg/m^3), $\mathbf{u}$ the velocity (m/s), E the total energy per mass $(m^2.s^{-2})$, p the pressure (N/m^2), T the temperature (K), μ the laminar dynamic viscosity $(kg/(m.s))$ and λ the laminar conductivity :

$$\lambda = \frac{\mu \, c_p}{P_r},$$

with the Prandtl laminar number $P_r = 0.72$ for typical atmospheric condition and c_p is the specific heat at constant pressure. $\mathcal{T}$ is the laminar stress tensor

$$\mathcal{T} = \mu \left(\nabla \mathbf{u} + {}^t\nabla \mathbf{u} - \frac{2}{3} (\operatorname{div} \mathbf{u}) \mathbb{I}_3 \right).$$

For the sake of notation and precision, let's denote $H = E + \frac{p}{\rho}$ the enthalpy and recall that we have $E = c_v T + \frac{1}{2}|\mathbf{u}|^2$, where c_v is the specific heat at constant volume and $p = \rho(\gamma - 1)\left(E - \frac{1}{2}|\mathbf{u}|^2\right)$, where $\gamma = c_p/c_v$ is the specific heat ratio.

2 Spatial Discretization

2.1 Space discretization

How can we talk about discretization without first talking about the notion of mesh. In the continuous case, we wish to solve the Navier-Stokes equations on a bounded open space $\Omega \subset \mathbb{R}^d$, d being the dimension. The discretization of this problem begins with the discretization of our computational domain. For simplicity's sake, let's consider $\Omega_h \subset \Omega$ a polygonal domain approximating our computational domain Ω. In our case, we consider a conformal tetrahedralization $\mathcal{T}_h$ of Ω_h, where in 2D the elements of our mesh are triangles and in 3D they are tetrahedra.

In this work, all our computations will be performed on unstructured and anisotropic meshes. An anisotropic mesh is a mesh whose elements can be stretched in a given direction, as opposed to an isotropic mesh whose elements are as regular (equilateral) as possible. For a simple example, see Figure 2.1 .

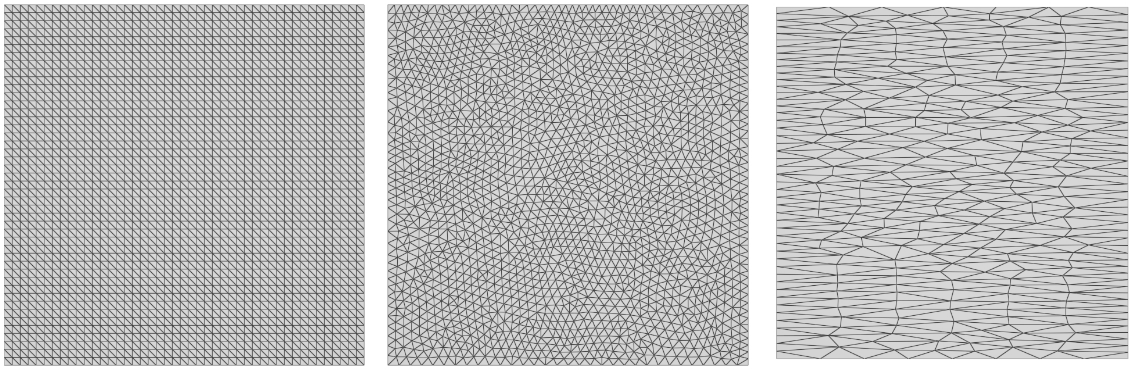


Figure 2.1 – Structured mesh on the left, isotropic unstructured mesh in the middle and anisotropic unstructured mesh on the right.

Because of the discretization method chosen for our equations, i.e. a mixed Element-finite/Volume-finite formulation, we construct a dual mesh from our triangulation/tetrahedrization. Our cells are reconstructed using the median method. Around each node i, a cell C_i is reconstructed by dividing each triangle/tetrahedron containing this node into 3 quadrilaterals in 2D or 4 hexahedrons in 3D.

In 2D, for each triangle, quadrilaterals are constructed by joining the vertex i with the triangle's center of gravity and the midpoint of the two edges coming from i.
In 3D, for each tetrahedron, hexahedrons are constructed by joining the vertex i with the center of gravity of the tetrahedron, the three centers of gravity of the faces and the midpoints of the three edges coming from i.

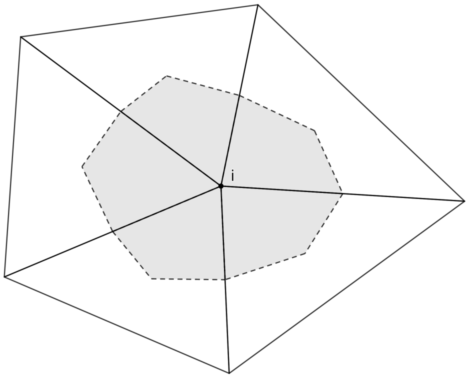


FIGURE 2.2 – Illustration of the construction of a dual cell at a vertex i of the mesh.

2.2 Mixed finite-Element/finite-Volume formulation

Now we are interested in the spatial discretization of the Navier Stokes system. The governing equations are discretized in space using a mixed finite-volume/finite-element method, finite-volumes are used for the convective terms, the diffusion terms are discretized using P1 Galerkin finite-elements on the tetrahedra. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain and $\mathcal{T}_h = \{K\}$ its conformal tetrahedrization. The adopted scheme is vertex-centered and at each node i of the mesh we build a dual cell C_i such that

$$\Omega = \bigcup_{i=1}^{\mathcal{N}} C_i \,,$$

We consider the Navier-Stokes system, written in condensed form,

$$\begin{cases} \partial_t W + \mathrm{div} F^C(W) = \mathrm{div} F^D(W), & (t,\mathbf{x}) \in [0,+\infty[\times\Omega, \\ W(0,\mathbf{x}) = W_0(\mathbf{x}), & \mathbf{x} \in \Omega, \\ W(t,\mathbf{x}) = \Phi_B(t,\mathbf{x}), & (t,\mathbf{x}) \in [0,+\infty[\times\partial\Omega. \end{cases} \tag{2.1}$$

with

$$\partial_t W + \mathrm{div} F^C(W) - \mathrm{div} F^D(W) = \partial_t W + \sum_{j=1}^{3} \partial_{x_j} F_j^C(W) - \sum_{j=1}^{3} \partial_{x_j} F_j^D(W) = 0, \tag{2.2}$$

where, with the following notation $\mathbf{u} = (u_1, u_2, u_3)$,

$$W = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{pmatrix}, \quad F_j^C(W) = \begin{pmatrix} \rho u_j \\ \rho u_1 u_j + p\,\delta_{1j} \\ \rho u_2 u_j + p\,\delta_{2j} \\ \rho u_3 u_j + p\,\delta_{3j} \\ (\rho E + p) u_j \end{pmatrix}, \quad F_j^D(W) = \begin{pmatrix} 0 \\ \mathcal{T}_{1j} \\ \mathcal{T}_{2j} \\ \mathcal{T}_{3j} \\ \sum_{i=1}^{3} u_i \mathcal{T}_{ij} + \lambda\, \partial_{x_j} T \end{pmatrix},$$

with

$$\mathcal{T}_{ij} = \mu \left(\partial_{x_j} u_i + \partial_{x_i} u_j - \frac{2}{3} \sum_{k=1}^{3} \partial_{x_k} u_k \delta_{ij} \right) .$$

The equations are integrated on each cell C_i and using the Stokes formula we have (omitting the boundary conditions)

$$\frac{d}{dt} \int_{C_i} W \, \mathrm{d}\mathbf{x} + \int_{\partial C_i} F^C(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma = \int_{\partial C_i} F^D(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma,$$

$\mathbf{n}_i$ is the outer normal to the cell surface ∂C_i. If we denote by $\mathcal{V}(i)$ the set of vertices j directly neighboring i, we can write

$$\begin{aligned} \int_{\partial C_i} F^C(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma &= \sum_{j \in \mathcal{V}(i)} \int_{\partial C_i \cap \partial C_j} F^C(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma \, , \\ \int_{\partial C_i} F^D(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma &= \sum_{j \in \mathcal{V}(i)} \int_{\partial C_i \cap \partial C_j} F^D(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma \, . \end{aligned}$$

For the convective fluxes we consider, with $\mathbf{n}_{ij} = \int_{\partial C_i \cap \partial C_j} \mathbf{n}_i \, \mathrm{d}\sigma$, the following approximation

$$\int_{\partial C_i \cap \partial C_j} F^C(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma \approx F^C_{|\partial C_i \cap \partial C_j} \cdot \mathbf{n}_{ij} \, .$$

The viscous terms will be discretized by Finite Element, so we will show an equivalence between our Finite Volume formulation and a Finite Element formulation. Let φ_i be the P_1 Finite Element basis function associated with the vertex p_i, we have (see Appendix for a proof)

$$\int_K \nabla \varphi_i \, \mathrm{d}\mathbf{x} = - \int_{\partial C_i \cap K} \mathbf{n} \, \mathrm{d}\sigma \, .$$

We will see that thanks to this formula we will be able to establish an equivalence between the Finite Volumes and the Finite Elements. Consider $T(i)$ the set of elements $K_i = (p_i, p_j, p_k, p_l)$ which have for common vertex p_i, then

$$\begin{aligned} \sum_{j \in \mathcal{V}(i)} \int_{\partial C_i \cap \partial C_j} F^D(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma &= \sum_{K_i \in T(i)} \int_{\partial C_i \cap K_i} F^D(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma \\ &= \sum_{K_i \in T(i)} F^D(W)_{|K_i} \cdot \int_{\partial C_i \cap K_i} \mathbf{n}_i \, \mathrm{d}\sigma \\ &= - \sum_{K_i \in T(i)} \int_{K_i} F^D(W)_{|K_i} \cdot \nabla \varphi_i \, \mathrm{d}\mathbf{x} \, . \end{aligned}$$

We will note respectively $\Phi_{ij}^C(W_i, W_j, \mathbf{n}_{ij})$ and $\Phi_i^D|_{K_i}(W_i, W_j, W_k, W_l)$ the approximations of $\int_{\partial C_i \cap \partial C_j} F^C(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma$ and $\int_{K_i} F^D(W)_{|K_i} \cdot \nabla \varphi_i \, \mathbf{dx}$. So we have the following semi-discrete scheme,

$$\frac{dW_i}{dt} = \frac{1}{|C_i|} \left(\sum_{j \in \mathcal{V}(i)} \Phi_{ij}^C(W_i, W_j, \boldsymbol{n}_{ij}) + \sum_{K_i \in T(i)} \Phi_i^D|_{K_i}(W_i, W_j, W_k, W_l) \right).$$

where W_i stands for the following quantity

$$W_i = \frac{1}{|C_i|} \int_{C_i} W \, \mathbf{dx}.$$

As described before, we have, with φ_i the P_1 Finite Element basis function associated with vertex p_i,

$$\Phi_i^D|_{K_i}(W_i, W_j, W_k, W_l) = -\int_{K_i} F^D(W)_{|K_i} \cdot \nabla \varphi_i \, \mathbf{dx},$$

where $F^D(W)_{|K_i}$ corresponds to the average value over the element K_i

$$F^D(W)_{|K_i} = \frac{F^D(W_i) + F^D(W_j) + F^D(W_k) + F^D(W_l)}{4}.$$

For the convective fluxes, the Riemann solver is either the Roe solver or the HLLC solver, which are described in the following subsections

$$\Phi^C(W_i, W_j, \mathbf{n}_{ij}) = \begin{cases} \Phi^{\text{Roe}}(W_i, W_j, \mathbf{n}_{ij}), \\ \text{or} \\ \Phi^{\text{HLLC}}(W_i, W_j, \mathbf{n}_{ij}). \end{cases}$$

2.3 Roe approximate Riemann solver

The Roe method consists in replacing the "exact" Riemann problem by a linearised Riemann problem, and solving the following problem at the interface $\partial C_i \cap \partial C_j$

$$\begin{cases} \partial_t W + \sum_{l=1}^{3} A_\ell(W) \partial_{x_\ell} W, \\ W(0, \mathbf{x}) = \begin{cases} W_i \text{ if } \mathbf{x} \in C_i, \\ W_j \text{ if } \mathbf{x} \in C_j. \end{cases} \end{cases}$$

So to describe the Roe scheme, we start by rewriting our system as follows

$$\partial_t W + \mathrm{div} F^C(W) = \partial_t W + \sum_{\ell=1}^{3} \partial_{x_\ell} F_\ell^C(W) = \partial_t W + \sum_{\ell=1}^{3} \nabla_W F_\ell^C(W) \partial_{x_\ell} W,$$

and let us note $A_\ell(W) = \nabla_W F_\ell^C(W)$ the Jacobian matrix of our flow, where for a fixed ℓ this matrix is given by

$$A_\ell(W) = \begin{pmatrix} 0 & \delta_{1\ell} & \delta_{2\ell} & \delta_{3\ell} & 0 \\ \left((\gamma-1)H - c^2\right)\delta_{1\ell} - u_1 u_\ell & u_\ell + (2-\gamma) u_1 \delta_{1\ell} & u_1 \delta_{2\ell} - (\gamma-1) u_2 \delta_{1\ell} & u_1 \delta_{3\ell} - (\gamma-1) u_3 \delta_{1\ell} & (\gamma-1)\delta_{1\ell} \\ \left((\gamma-1)H - c^2\right)\delta_{2\ell} - u_2 u_\ell & u_2 \delta_{1\ell} - (\gamma-1) u_1 \delta_{2\ell} & u_\ell + (2-\gamma) u_2 \delta_{2\ell} & u_2 \delta_{3\ell} - (\gamma-1) u_3 \delta_{2\ell} & (\gamma-1)\delta_{2\ell} \\ \left((\gamma-1)H - c^2\right)\delta_{3\ell} - u_3 u_\ell & u_3 \delta_{1\ell} - (\gamma-1) u_1 \delta_{3\ell} & u_3 \delta_{2\ell} - (\gamma-1) u_2 \delta_{3\ell} & u_\ell + (2-\gamma) u_3 \delta_{3\ell} & (\gamma-1)\delta_{3\ell} \\ u_\ell\left((\gamma-2)H - c^2\right) & H\delta_{1\ell} + (1-\gamma) u_1 u_\ell & H\delta_{2\ell} + (1-\gamma) u_2 u_\ell & H\delta_{3\ell} + (1-\gamma) u_3 u_\ell & \gamma u_\ell \end{pmatrix}.$$

Let's note $\lambda_\ell^{(n)}$ the n-th eigenvalue of the matrix $A_\ell(W)$, then for $1 \leqslant n \leqslant 5$ they are given by

$$\lambda_\ell^{(1)}(W) = u_\ell - c,\;\; \lambda_\ell^{(2)}(W) = \lambda_\ell^{(3)}(W) = \lambda_\ell^{(4)}(W) = u_\ell,\;\; \lambda_\ell^{(5)}(W) = u_\ell + c\,,$$

and the corresponding eigenvectors are

$$r_\ell^1(W) = \begin{pmatrix} 1 \\ u_1 - c\,\delta_{1\ell} \\ u_2 - c\,\delta_{2\ell} \\ u_3 - c\,\delta_{3\ell} \\ H - u_\ell c \end{pmatrix},\; r_\ell^2(W) = \begin{pmatrix} \delta_{1\ell} \\ u_1\,\delta_{1\ell} + \delta_{2\ell} + \delta_{3\ell} \\ u_2\,\delta_{1\ell} \\ u_3\,\delta_{1\ell} \\ \frac{1}{2}|u|^2\,\delta_{1\ell} + (\delta_{2\ell} + \delta_{3\ell}) u_1 \end{pmatrix},\; r_\ell^3(W) = \begin{pmatrix} \delta_{2\ell} \\ u_1\,\delta_{2\ell} \\ u_2\,\delta_{2\ell} + \delta_{1\ell} + \delta_{3\ell} \\ u_3\,\delta_{2\ell} \\ \frac{1}{2}|u|^2\,\delta_{2\ell} + (\delta_{1\ell} + \delta_{3\ell}) u_2 \end{pmatrix},$$

$$r_\ell^4(W) = \begin{pmatrix} \delta_{3\ell} \\ u_1\,\delta_{3\ell} \\ u_2\,\delta_{3\ell} \\ u_3\,\delta_{3\ell} + \delta_{1\ell} + \delta_{2\ell} \\ \frac{1}{2}|u|^2\,\delta_{3\ell} + (\delta_{1\ell} + \delta_{2\ell}) u_3 \end{pmatrix},\; r_\ell^5(W) = \begin{pmatrix} 1 \\ u_1 + c\,\delta_{1\ell} \\ u_2 + c\,\delta_{2\ell} \\ u_3 + c\,\delta_{3\ell} \\ H + u_\ell c \end{pmatrix}.$$

If we set $\overline{W}_{ij} = \left(\rho_i^{\frac{1}{2}} W_i + \rho_j^{\frac{1}{2}} W_j\right) / \left(\rho_i^{\frac{1}{2}} + \rho_j^{\frac{1}{2}}\right)$ then the matrices A_ℓ satisfy the following jump condition

$$F_\ell^C(W_i) - F_\ell^C(W_j) = A_\ell(\overline{W}_{ij})(W_i - W_j).$$

Then the Roe scheme is given by

$$\Phi^{\text{Roe}}(W_i, W_j, \mathbf{n}_{ij}) = \frac{1}{2}\left(F_\ell^C(W_i) + F_\ell^C(W_j)\right) \cdot \mathbf{n}_{ij} + \frac{1}{2}\left|A_\ell(\overline{W}_{ij})\right|(W_i - W_j).$$

In the following, we'll mention a parameter $\gamma \in]0,1]$ that we'll specify ; this is a coefficient introduced to control the upwind term in the numerical flux, because the diagramRoe scheme we're considering is actually written as

$$\Phi^{\text{Roe}}(W_i, W_j, \mathbf{n}_{ij}) = \frac{1}{2}\left(F_\ell^C(W_i) + F_\ell^C(W_j)\right) \cdot \mathbf{n}_{ij} + \frac{\gamma}{2}\left|A_\ell(\overline{W}_{ij})\right|(W_i - W_j).$$

2.4 HLLC approximate Riemann solver

We consider for our study the HLLC (Harten-Lax-van-Leer contact) solver. The idea is to consider locally a simplified Riemann problem with two intermediate states depending on the local left and right states. The simplified solution to the Riemann problem consists of a contact wave with a velocity S_M and two acoustic waves, which may be either shocks or expansion fans. The acoustic waves have the smallest and the largest velocities (S_i and S_j, respectively) of all the waves present in the exact solution. If $S_i > 0$ then the flow is supersonic from left to right and the upwind flux is simply defined from $F(W_i)$ where W_i is the state to the left of the discontinuity. Similarly, if $S_j < 0$ then the flow is supersonic from right to left and the flux is defined from $F(W_j)$ where W_j is the state to the right of the discontinuity. In the more difficult subsonic case when $S_i < 0 < S_j$ we have to calculate $F(W_i^*)$ or $F(W_j^*)$. Consequently, the HLLC flux is given by

$$\Phi^{\text{HLLC}}(W_i, W_j, \mathbf{n}_{ij}) = \begin{cases} F(W_i) \cdot \mathbf{n}_{ij} & \text{if } S_i > 0, \\ F(W_i^*) \cdot \mathbf{n}_{ij} & \text{if } S_i \leqslant 0 < S_M, \\ F(W_j^*) \cdot \mathbf{n}_{ij} & \text{if } S_M \leqslant 0 \leqslant S_j, \\ F(W_j) \cdot \mathbf{n}_{ij} & \text{if } S_j < 0. \end{cases}$$

where W_i^* and W_j^* are evaluated as follows. Let us denote $\eta = \mathbf{u} \cdot \mathbf{n}$, the following evaluations are proposed

$$W^* = \frac{1}{S - S_M} \begin{pmatrix} \rho(S-\eta) \\ \rho\mathbf{u}(S-\eta) + (p^* - p)\mathbf{n} \\ \rho E(S-\eta) + p^* S_M - p\eta \end{pmatrix},$$

with $p^* = \rho(S-\eta)(S_M - \eta) + p$. For the contact wave we consider :

$$S_M = \frac{\rho_j \eta_j (S_j - \eta_j) - \rho_i \eta_i (S_i - \eta_i) + p_i - p_j}{\rho_j (S_j - \eta_j) - \rho_i (S_i - \eta_i)}$$

and the acoustic wave speeds based on the Roe average $(\bar{.})$:

$$S_i = \min(\eta_i - c_i, \bar{\eta} - \bar{c}), \quad S_j = \max(\eta_j + c_j, \bar{\eta} + \bar{c}).$$

2.5 Boundary conditions

No-slip Boundary Condition. For no-slip boundary conditions, $\mathbf{u} = 0$ is trongly enforced at each iteration. Consistently, we impose $\Phi_\rho = 0$ and $\Phi_{\rho\mathbf{u}} = 0$ at the boundary. The energy flux is fixed according to the desired temperature behavior.

Slip Boundary Condition. For this boundary condition we impose weakly

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

If this condition is satisfied then

$$\Phi_{Slip} = F^C(W) \cdot \mathbf{n} = \begin{pmatrix} 0 \\ p\,\mathbf{n} \\ 0 \end{pmatrix}.$$

Far-field Boundary Condition. Far field boundary conditions are imposed weakly. The far-field state W_∞ is defined by the flow conditions of the problem at hand. The flux at the far-field boundary is computed using the Roe or the HLLC approximate Riemann solver :

$$\Phi_{farfield} = \Phi^{\text{Roe/HLLC}}(W_i, W_\infty, \mathbf{n}_{ij}).$$

3 MUSCL scheme

The above schemes with Roe or HLLC are spatially first-order accurate and they can be extended to second order by applying the MUSCL method. Van Leer proposed [228, 227] [1] [2] 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 extrema. The resulting MUSCL method produces positive second-order schemes. We describe now an extension of MUSCL to unstructured triangulations with dual cells.

We consider extrapolated values of W_{ij} and W_{ji} of W at the interface between two cells C_i and C_j (see Figure 2.3) in order to increase the order of accuracy of the scheme. the numerical flux becomes :

$$\Phi^C(W_{ij}, W_{ji}, \mathbf{n}_{ij}).$$

With values at the interface inspired by the Taylor expansion of order 1 ,

$$W_{ij} = W_i + \frac{1}{2}(\nabla W)_{ij} \cdot \vec{ij},$$
$$W_{ji} = W_j + \frac{1}{2}(\nabla W)_{ji} \cdot \vec{ij}.$$

---

[1] [227] B. Van Leer. Towards the Ultimate Conservative Difference Scheme IV : A new approach to numerical convection. J. Comp. Phys., 23 :276–299, 1977.

[2] [228] B. Van Leer. Towards the Ultimate Conservative Difference Scheme V : A Second Order Sequel to Godunov's Method. J. Comp. Phys., 32 :101–136, 1979.

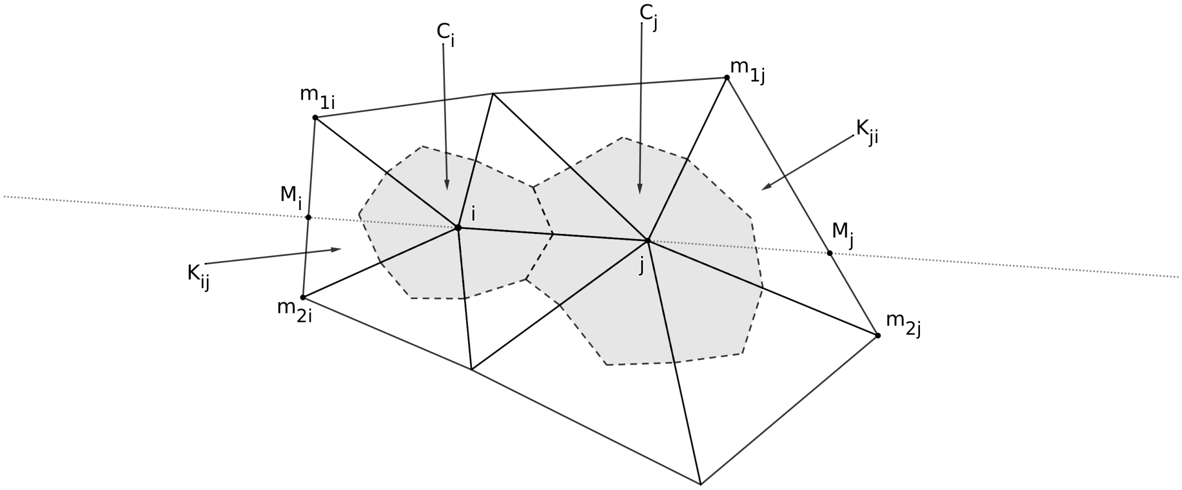


FIGURE 2.3 – Two adjacent dual cells.

3.1 V4 scheme

With this scheme, the gradients $(\nabla W)_{ij}$ and $(\nabla W)_{ji}$ are expressed by combining the centered and off-center gradients according to the following form :

$$(\nabla W)_{ij} \cdot \vec{ij} = (1-\beta)(\nabla W)^C_{ij} \cdot \vec{ij} + \beta(\nabla W)^U_{ij} \cdot \vec{ij},$$
$$(\nabla W)_{ji} \cdot \vec{ij} = (1-\beta)(\nabla W)^C_{ij} \cdot \vec{ij} + \beta(\nabla W)^D_{ij} \cdot \vec{ij},$$

where the centered gradients are constructed as follows :

$$(\nabla W)^C_{ij} \cdot \vec{ij} = W_j - W_i \,.$$

Let φ_i be the $P1$ Finite Element basis function associated with the vertex i, and let's denote K_{ij} and K_{ji} the upstream and downstream elements respectively, then the upstream gradient $(\nabla W)^U_{ij} \cdot \vec{ij}$ and the downstream gradient $(\nabla W)^D_{ij} \cdot \vec{ij}$ are defined

$$(\nabla W)^D_{ij} \cdot \vec{ij} = \sum_{k \in K_{ij}} W_k \nabla \varphi_{k|K_{ij}},$$

and

$$(\nabla W)^U_{ij} \cdot \vec{ij} = \sum_{k \in K_{ji}} W_k \nabla \varphi_{k|K_{ji}},$$

β is a parameter controlling the amount of upwinding with $\beta \in [0,1]$. The most accurate β-scheme is obtained for $\beta = \frac{1}{3}$ [3] [4] . Indeed, it can be demonstrated that this scheme is third-order for the two-dimensional linear advection equation on structured triangular meshes and, on unstructured meshes, a second-order scheme with a fourth-order numerical dissipation is obtained.

3.2 V6 scheme

This scheme introduces, in its gradient estimation, a new gradient called the average nodal gradient on cell C_i, denoted $(\nabla W)_i$, and corresponding to the average of the gradients on all element K containing node i

$$(\nabla W)_i = \frac{1}{|C_i|} \sum_{K \in C_i} \frac{|T|}{4} \sum_{k \in K} W_k \nabla \varphi_{k|K} \, .$$

The evaluation of gradients for solution reconstruction at cell interfaces is then

$$\begin{aligned}(\nabla W)_{ij} \cdot \vec{ij} = {} & (1-\beta)(\nabla W)^C_{ij} \cdot \vec{ij} + \beta (\nabla W)^U_{ij} \cdot \vec{ij} \\ & + \alpha_c \left[(\nabla W)^U_{ij} \cdot \vec{ij} - 2(\nabla W)^C_{ij} \cdot \vec{ij} + (\nabla W)^D_{ij} \cdot \vec{ij} \right] \\ & + \alpha_d \left[(\nabla W)_{M_i} \cdot \vec{ij} - 2(\nabla W)_i \cdot \vec{ij} + (\nabla W)_j \cdot \vec{ij} \right] ,\end{aligned}$$

and

$$\begin{aligned}(\nabla W)_{ji} \cdot \vec{ij} = {} & (1-\beta)(\nabla W)^C_{ji} \cdot \vec{ij} + \beta (\nabla W)^U_{ji} \cdot \vec{ij} \\ & + \alpha_c \left[(\nabla W)^U_{ji} \cdot \vec{ij} - 2(\nabla W)^C_{ji} \cdot \vec{ij} + (\nabla W)^D_{ji} \cdot \vec{ij} \right] \\ & + \alpha_d \left[(\nabla W)_{M_j} \cdot \vec{ij} - 2(\nabla W)_i \cdot \vec{ij} + (\nabla W)_j \cdot \vec{ij} \right] ,\end{aligned}$$

where M_i is the point of intersection of the straight line passing through nodes i and j with the face opposite node i of the element K_{ij}, and M_j is the point of intersection of the straight line passing through nodes i and j with the face opposite node j of the element K_{ji} (see Figure 2.3 for a two-dimensional example). For pararmeter values, we choose $\beta = \frac{1}{3}$, $\alpha_c = -\frac{1}{30}$ and $\alpha_d = -\frac{2}{15}$ and the V6 scheme is obtained for these values.

4 Temporal discretization

As we shall see later, in this thesis we are mainly interested in unsteady turbulent flows. These flows are very time-consuming and it is preferable to have large time steps,

---

[3] B. Koren, A robust upwind discretization method for advection diffusion and source terms, Notes Numer. Fluid Mech. 45 (1993) 117–138.

[4] C. Debiez, A. Dervieux, Mixed-element-volume MUSCL methods with weak viscosity for steady and unsteady flow calculations, Comput. Fluids 29 (2000) 89–118.

so only implicit Backward Differentiation Formulae (BDF) schemes will be used. For simplicity's sake, let us rewrite the semi-discrete scheme as follows

$$\frac{dW_i(t)}{dt} = \frac{1}{|C_i|}\,\Phi_i(W(t)),$$

then the 1st-order scheme, BDF1, is simply written

$$W_i^{n+1} - W_i^n = \frac{\Delta t^n}{|C_i|}\,\Phi_i(W^{n+1}),$$

and the 2nd order scheme, BDF2, is given by

$$\alpha_{n+1} W_i^{n+1} + \alpha_n W_i^n + \alpha_{n-1} W_i^{n-1} = \frac{\Delta t^n}{|C_i|}\,\Phi_i(W^{n+1}),$$

with

$$\alpha_{n+1} = \frac{1+2\tau}{1+\tau}, \quad \alpha_n = -1-\tau, \quad \alpha_{n-1} = \frac{\tau^2}{1+\tau}, \quad \tau = \frac{\Delta t^n}{\Delta t^{n+1}}.$$

5 Appendix : P_1-FEM/Median-FVM compatibility

The purpose of this section is to show that the median finite volume is close to the P_1 finite element, with the consequences that (1) both method can be combined in a rather accurate MUSCL-FEM approximation, and (2) this approximation can be analysed like FEM when anisotropic error analysis is applied for adapting the mesh. We shall show in this appendix the following relationship, which expresses the equivalence between both approximations :

$$\int_K \nabla\varphi_i \,\mathrm{d}x = -\int_{\partial C_i \cap K} \mathbf{n}\,\mathrm{d}\sigma\,.$$

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with $d = 2, 3$. Let $\mathcal{T}_h$ be a conforming triangulation or tetrahedralization of Ω, and $K \in \mathcal{T}_h$ a genetic element. For each vertex i we build a dual cell C_i according to the median rule.

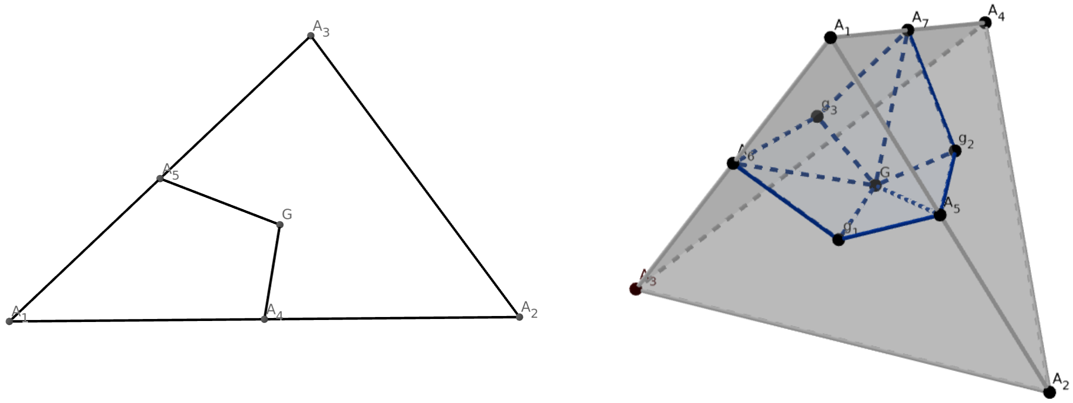


FIGURE 2.4 – 2D and 3D element with dual cell.

Notations :

- In $2D$:
  - K = $A_1A_2A_3$
  - $\partial K = A_1A_2 \cup A_1A_3 \cup A_2A_3$
  - $\partial C_i \cap K = A_4G \cup A_5G$
  - $C_i \cap \partial K = A_1A_4 \cup A_1A_5$

- In $3D$:
  - K = $A_1A_2A_3A_4$
  - $\partial K = A_1A_2A_3 \cup A_1A_2A_4 \cup A_1A_3A_4 \cup A_2A_3A_4$
  - $\partial C_i \cap K = A_6g_1Gg_3 \cup A_5g_1Gg_2 \cup A_7g_2Gg_3$
  - $C_i \cap \partial K = A_1A_6g_1A_5 \cup A_1A_5g_2A_7 \cup A_1A_7g_3A_6$

First, let's consider a matrix $\mathbf{\Psi}$ of size 3×3. Using Stokes' formula we have

$$\int_{C_i \cap K} \operatorname{div} \mathbf{\Psi} \, \mathrm{d}\mathbf{x} = \int_{\partial C_i \cap K} \mathbf{\Psi} \, \mathbf{n} \, \mathrm{d}\sigma + \int_{C_i \cap \partial K} \mathbf{\Psi} \, \mathbf{n} \, \mathrm{d}\sigma,$$

so, taking $\Psi = \mathbb{I}_3$

$$-\int_{\partial C_i \cap K} \mathbf{n}\, \mathrm{d}\sigma = \int_{C_i \cap \partial K} \mathbf{n}\, \mathrm{d}\sigma = \mathbf{n}_{|C_i \cap \partial K}\, |C_i \cap \partial K|.$$

Furthermore, we have

$$\int_K \nabla \varphi_i \, \mathbf{dx} = \int_{\partial K} \mathbf{n}\, \varphi_i \, \mathbf{dx}.$$

● In dimension 2 :

$$\begin{aligned}\int_{\partial K} \mathbf{n}\, \varphi_i \, \mathbf{dx} &= \int_{A_1A_2} \mathbf{n}\, \varphi_i \, \mathbf{dx} + \int_{A_1A_3} \mathbf{n}\, \varphi_i \, \mathbf{dx} + \int_{A_2A_3} \mathbf{n}\, \varphi_i \, \mathbf{dx},\\ &= \mathbf{n}_{|A_1A_2} \int_{A_1A_2} \varphi_i \, \mathbf{dx} + \mathbf{n}_{|A_1A_3} \int_{A_1A_3} \varphi_i \, \mathbf{dx} + \mathbf{n}_{|A_2A_3} \int_{A_2A_3} \varphi_i \, \mathbf{dx},\end{aligned}$$

in $2D$ the basis functions $\mathbb{P}_1$ are the hat functions which are equal to 1 on their respective vertex and 0 on adjacent vertices. Thus, on an edge, the calculation of the integral can be reduced to the calculation of the area of a right-angled triangle of height 1, with a base of $|A_1A_2|$ or $|A_1A_3|$. Note that φ_i is zero on A_2A_3. We have

$$\begin{aligned}\int_{\partial K} \mathbf{n}\, \varphi_i \, \mathbf{dx} &= \frac{1}{2}\, \mathbf{n}_{|A_1A_2}\, |A_1A_2| + \frac{1}{2}\, \mathbf{n}_{|A_1A_3}\, |A_1A_3|,\\ &= \mathbf{n}_{|C_i \cap \partial K}\, |C_i \cap \partial K|,\end{aligned}$$

car $\frac{1}{2}|A_1A_2| = |A_1A_4|$ et $\frac{1}{2}|A_1A_3| = |A_1A_5|$.

● In dimension 3 :

$$\begin{aligned}\int_{\partial K} \mathbf{n}\, \varphi_i \, \mathbf{dx} &= \int_{A_1A_2A_3} \mathbf{n}\, \varphi_i \, \mathbf{dx} + \int_{A_1A_3A_4} \mathbf{n}\, \varphi_i \, \mathbf{dx} + \int_{A_1A_2A_4} \mathbf{n}\, \varphi_i \, \mathbf{dx} + \int_{A_2A_3A_4} \mathbf{n}\, \varphi_i \, \mathbf{dx},\\ &= \mathbf{n}_{|A_1A_2A_3} \int_{A_1A_2A_3} \varphi_i \, \mathbf{dx} + \mathbf{n}_{|A_1A_3A_4} \int_{A_1A_3A_4} \varphi_i \, \mathbf{dx} + \mathbf{n}_{|A_1A_2A_4} \int_{A_1A_2A_4} \varphi_i \, \mathbf{dx}\\ &\quad + \mathbf{n}_{|A_2A_3A_4} \int_{A_2A_3A_4} \varphi_i \, \mathbf{dx},\end{aligned}$$

we have the same arguments as in $2D$, this time the integral over a triangle is reduced to the calculation of the volume of a tetrahedron of height 1 and base equal to $|A_1A_2A_3|$, $|A_1A_3A_4|$ or $|A_1A_2A_4|$, φ_i is zero $A_2A_3A_4$. So

$$\begin{aligned}\int_{\partial K} \mathbf{n}\, \varphi_i \, \mathbf{dx} &= \frac{1}{3}\, \mathbf{n}_{|A_1A_2A_3}\, |A_1A_2A_3| + \frac{1}{3}\, \mathbf{n}_{|A_1A_3A_4}\, |A_1A_3A_4| + \frac{1}{3}\, \mathbf{n}_{|A_1A_2A_4}\, |A_1A_2A_4|,\\ &= \mathbf{n}_{|C_i \cap \partial K}\, |C_i \cap \partial K|,\end{aligned}$$

because $|A_1A_5g_1A_6| = \frac{1}{3}|A_1A_2A_3|$, $|A_1A_5g_2A_7| = \frac{1}{3}|A_1A_2A_4|$ and $|A_1A_6g_3A_7| = \frac{1}{3}|A_1A_3A_4|$.

$\square$

CHAPITRE 3

Turbulence modelling

1 Introduction

Turbulence refers to the chaotic nature of a flow, with seemingly random variations in space and time. Turbulent flows occur when the viscosity forces that the fluid opposes to move become relatively weak compared to the source of kinetic energy that sets the fluid in motion.

Turbulent flow is also characterized by the presence of structures of very different sizes that interact with each other, increasing the rate of energy dissipation and giving the flow a dissipative character. Large eddies, which contain most of the turbulent kinetic energy, are unstable. They split by transferring their energy to smaller structures. The energy carried by large structures is successively transferred to smaller and smaller structures. This transfer of energy from larger structures to smaller ones is known as the energy cascade.

2 To model or not to model (the turbulence)

In this chapter we will present different ways of modelling turbulence for numerical simulation, but first we can ask why we need to model turbulence.

The simplest way to describe a flow is to solve the Navier-Stokes equations directly, without introducing any modelling. We would then need to solve all the scales of the flow numerically, including the smallest, which are responsible for energy dissipation. Numerical resolution then requires a sufficiently fine mesh, so we speak of direct numerical simulation (DNS).

The number of nodes required for DNS can be quantified. For homogeneous isotropic turbulence, the number of nodes required is of the order of $Re^{(3^{d-1})/4}$ in each direction, where d denotes the spatial dimension and is either 2 or 3. This illusttrates that spatial discretisation requires the use of very fine meshes as the Reynolds number increases. Think that the flow around a commercial aircraft can be at Reynolds 10^8. DNS may be well suited to tackling certain fundamental research problems, but is impractical for most industrial problems.

Various modelling approaches have been developed to resolve the constraints associated with DNS. In this thesis, we will look at two of them in particular : models of the

Reynolds averaged Navier-Stokes equations (RANS) and large-eddy simulation (LES).

3 Turbulence models

In this section we will present some turbulence models that we will use later.

3.1 Reynolds-Averaged Navier-Stokes (RANS)

RANS approaches involve modelling all turbulent scales using a statistical approach. The Navier-Stokes equations are averaged by decomposing, using the Reynolds decomposition, each of the physical variables of the flow into an average part and a fluctuating part :

$$\phi = \overline{\phi} + \phi', \tag{3.1}$$

where $\overline{\phi}$ is the mean of ϕ and ϕ' is the fluctuation of ϕ. Let's look at our average operator. There are four mathematical rules that this operator must satisfy, which we will call "Reynolds axioms" in reference to J. Kampé de Fériet in [37] :

- $\overline{\phi + \psi} = \overline{\phi} + \overline{\psi}$
- $\overline{a\phi} = a\overline{\phi}$, a constant
- $\overline{\overline{\phi}\psi} = \overline{\phi}\overline{\psi}$
- $\overline{\lim_n \phi_n} = \lim_n \overline{\phi_n}$

For a function depending only on one variable $\phi = \phi(t)$, take for example the time average, as suggested by Reynolds

$$\overline{\phi}(t) = \frac{1}{2T}\int_{t-T}^{t+T} \phi(t')\,\mathrm{d}t'.$$

This relation does not satisfy the third point of the axioms for $\psi = 1$, namely $\overline{\overline{\phi}} = \overline{\phi}$. We can try to avoid this problem by passing to the limit when $T \to \infty$. The Reynolds axioms are then satisfied, but only constants are found as mean values. Consider the ensemble mean of $\phi(\mathbf{x}, t)$:

$$\langle \phi \rangle(\mathbf{x}, t) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \phi_k(\mathbf{x}, t),$$

The same experiment is repeated N times, and the N realisations obtained at the end of time T are averaged. For a stationary random process we have

$$\langle\phi\rangle(\mathbf{x}) = \lim_{T\to\infty} \frac{1}{T}\int_0^T \phi(\mathbf{x},t)\,\mathrm{d}t.$$

The mean is taken to be the ensemble mean $\overline{\phi}(\mathbf{x},t) = \langle\phi\rangle(\mathbf{x},t)$. For the compressible Navier-Stokes equations, if we introduce the Reynolds decomposition directly we obtain an additional correlation term increasing the number of unknowns. To avoid this, Favre proposed in 1965 to use a mass-weighted average. This is called the Favre mean and is denoted $\widehat{\phi}$:

$$\phi = \widehat{\phi} + \phi''\,,$$

with

$$\begin{aligned} \widehat{\phi} &= \frac{\overline{\rho\phi}}{\overline{\rho}}, \\ \overline{\rho\phi''} &= \overline{\rho}\widehat{\phi''} = 0. \end{aligned}$$

The Reynolds-averaged Navier-Stokes equations obtained by applying the Favre decomposition are written as :

$$\begin{aligned} &\partial_t\overline{\rho} + \mathrm{div}\,(\overline{\rho}\widehat{\mathbf{u}}) = 0, \\ &\partial_t(\overline{\rho}\widehat{\mathbf{u}}) + \mathrm{div}\,(\overline{\rho}\,\widehat{\mathbf{u}}\otimes\widehat{\mathbf{u}}) + \nabla\overline{p} = \mathrm{div}\,\left(\widehat{\mathcal{T}} - \overline{\rho\mathbf{u}''\otimes\mathbf{u}''}\right), \\ &\partial_t(\overline{\rho}\widehat{E}) + \mathrm{div}\,\left(\left(\overline{\rho}\widehat{E} + \overline{p}\right)\widehat{\mathbf{u}}\right) = \mathrm{div}\,\left(\lambda\nabla\overline{T} - \overline{\rho C_p\mathbf{u}''T''}\right) + \mathrm{div}\,\left(\overline{\tau''\cdot\mathbf{u}''} - \overline{\rho\frac{\mathbf{u}''\cdot\mathbf{u}''}{2}\mathbf{u}''}\right) \\ &\qquad\qquad + \mathrm{div}\,\left(\left(\widehat{\mathcal{T}} - \overline{\rho\mathbf{u}''\otimes\mathbf{u}''}\right)\cdot\widehat{\mathbf{u}}\right). \end{aligned}$$

We have four additional terms, the Reynolds-stress tensor $\overline{\rho\mathbf{u}''\otimes\mathbf{u}''}$, the turbulent heat-flux vector $\overline{\rho C_p\mathbf{u}''T''}$, the molecular diffusion $\overline{\tau''\cdot\mathbf{u}''}$ and the turbulent transport $\overline{\rho\frac{\mathbf{u}''\cdot\mathbf{u}''}{2}\mathbf{u}''}$. These terms must be modeled so that the above equations can be solved !

For the Reynolds stress tensor, Boussinesq (1877) proposed to model it with a behavior law based on turbulent viscosity μ_t,

$$\mathcal{T}_R = -\overline{\rho\mathbf{u}''\otimes\mathbf{u}''} = \mu_t\left(\nabla\widehat{\mathbf{u}} + {}^t\nabla\widehat{\mathbf{u}} - \frac{2}{3}(\mathrm{div}\,\widehat{\mathbf{u}})\mathbb{I}_3\right) - \frac{2}{3}\overline{\rho}k\mathbb{I}_3,$$

a linear behavior law based on a turbulent Prandtl number P_{rt} is used to model the turbulent heat-flux from the temperature gradient

$$q_T = \overline{\rho C_p\mathbf{u}''T''} = -\frac{\mu_t C_p}{Pr_t}\nabla\overline{T}.$$

For the molecular diffusion and turbulent transport the most commonly used approximation is

$$\overline{\tau'' \cdot \mathbf{u}''} - \rho \overline{\frac{\mathbf{u}'' \cdot \mathbf{u}''}{2} \mathbf{u}''} = \left(\mu + \mu_t\right) \nabla k.$$

Finally, the Reynolds Averaged Navier-Stokes equations can be written as follows

$$\begin{cases} \partial_t \overline{\rho} + \operatorname{div}\left(\overline{\rho} \widehat{\mathbf{u}}\right) = 0, \\ \partial_t (\overline{\rho} \widehat{\mathbf{u}}) + \operatorname{div}\left(\overline{\rho}\, \widehat{\mathbf{u}} \otimes \widehat{\mathbf{u}}\right) + \nabla \overline{p} = \operatorname{div}\left(\left(1 + \frac{\mu_t}{\mu}\right) \widehat{\mathcal{T}}\right), \\ \partial_t (\overline{\rho} \widehat{E}) + \operatorname{div}\left(\left(\overline{\rho} \widehat{E} + \overline{p}\right) \widehat{\mathbf{u}}\right) = \operatorname{div}\left((\lambda + \lambda_t) \nabla \overline{T}\right) + \operatorname{div}\left(\left(\left(1 + \frac{\mu_t}{\mu}\right) \widehat{\mathcal{T}}\right) \cdot \widehat{\mathbf{u}}\right). \end{cases}$$

To close the system it is necessary to determine μ_t, to do this, we can model this term using a turbulence model. In this thesis, we will focus only on a so-called first-order model with one transport equation, the Spalart-Allmaras model.

Remark : The turbulent conductivity is deduced from the turbulent viscosity with $\frac{\mu_t C_p}{Pr_t}$ where Pr_t is assumed to be constant and equal to 0.9 .

3.1.1 Spalart-Allmaras model

As seen previously, in the case of the RANS approach the Navier-Stokes equations are completed by a turbulence model defined by one or more equation. The laminar dynamic viscosity μ is replaced in the equations by the sum between the laminar and the turbulent dynamic viscosity $\mu + \mu_t$, and the laminar conductivity λ is replaced by the sum of the laminar and the turbulent conductivity $\lambda + \lambda_t$. The turbulent dynamis viscosity μ_t is given by the turbulence model and the turbulent conductivity is expressed from the Prandtl turbulent number. Here we choose the following Spalart-Allmaras one equation turbulence model :

$$\partial_t(\rho\tilde{\nu}) + \operatorname{div}(\rho \mathbf{v} \tilde{\nu}) = \rho c_{b1} \tilde{S} \tilde{\nu} - \rho c_{w1} f_w \left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{\rho}{\sigma} \left(\operatorname{div}\left((\nu + \tilde{\nu}) \nabla \tilde{\nu}\right) + c_{b2} \|\nabla \tilde{\nu}\|^2\right).$$

The turbulent eddy viscosity is computed from :

$$\mu_t = \rho \tilde{\nu} f_{v1}, \quad \text{where} \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \quad \text{and} \quad \chi = \frac{\tilde{\nu}}{\nu}, \; \nu = \frac{\mu}{\rho}.$$

Additional definitions are given by the following equations :

$$\tilde{S} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \quad \text{where} \quad \Omega = \|\mathbf{rot}\, \mathbf{v}\|.$$

Symbol d holds for the distance from the field point to the nearest wall and

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}.$$

The constants are

$$\sigma = \frac{2}{3}, \quad c_{b1} = 0.1355, \quad c_{b2} = 0.622, \quad \kappa = 0.41, \quad c_{w2} = 0.3, \quad c_{w3} = 2, \quad c_{v1} = 7.1,$$
$$c_{w1} = \frac{c_{b1}}{\kappa} + \frac{1 + c_{b2}}{\sigma}.$$

Finally, the function f_w is computed as :

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{1/6} \quad \text{with} \;\; g = r + c_{w2}(r^6 - r) \;\; \text{and} \;\; r = \min \left(\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10 \right).$$

In our case we are interested in a version of the Spalart-Allmaras model with a correction term for the rotation. Spalart and Shur propose a simple modification of the one-equation transport turbulence model of Spalart and Allmaras [128, 123] , to make it more sensitive to rotational and curvature effects. It is just a simple modification of the original model, the production term $\rho c_{b1}\tilde{S}\tilde{\nu}$ is multiplied by the "rotation function" f_{r1} :

$$f_{r1}(r^*, \tilde{r}) = (1 + c_{r1}) \frac{2r^*}{1 + r^*} \left(1 - c_{r3} \tan^{-1}(c_{r2}\tilde{r}) \right) - c_{r1}.$$

The nondimensional quantities r^* and $\tilde{r}$ are given by

$$r^* = \frac{S}{\Omega},$$
$$\tilde{r} = 2 \frac{\Omega_{ik} S_{jk}}{D^4} \left(\frac{dS_{ij}}{dt} + \left(\varepsilon_{imn} S_{jn} + \varepsilon_{jmn} S_{in} \right) \omega_m \right),$$

where the constants are $c_{r1} = 1$, $c_{r2} = 12$, $c_{r3} = 1$ and

$$S_{ij} = \frac{1}{2} \left(\partial_{x_j} v_i + \partial_{x_i} v_j \right), \quad \Omega_{ij} = \frac{1}{2} \left(\left(\partial_{x_j} v_i - \partial_{x_i} v_j \right) + 2\varepsilon_{mji}\, \omega_m \right),$$
$$S^2 = 2 S_{ij} S_{ij}, \quad \Omega^2 = 2\, \Omega_{ij} \Omega_{ij}, \quad D^2 = \frac{1}{2} \left(S^2 + \Omega^2 \right).$$

3.1.2 Discretization of the Spalart-Allmaras model

For the discretization of our Spalart-Allmaras model we integrate our equation on each cell C_i, by noting $f^s = \rho c_{b1} \tilde{S} \tilde{\nu} - \rho c_{w1} f_w \left(\frac{\tilde{\nu}}{d} \right)^2$ and using the Stokes formula

$$\frac{d}{dt} \int_{C_i} \rho \tilde{\nu} \, \mathbf{dx} + \int_{\partial C_i} (\rho \mathbf{u} \tilde{\nu}) \cdot \mathbf{n}_i \, d\sigma = \int_{C_i} \frac{\rho}{\sigma} \left(\operatorname{div} \left((\nu + \tilde{\nu}) \nabla \tilde{\nu} \right) + c_{b2} \|\nabla \tilde{\nu}\|^2 \right) \mathbf{dx} + \int_{C_i} f^s \, \mathbf{dx}.$$

Now noticing that $\mathrm{div}(\tilde{\nu}\nabla\tilde{\nu}) = \|\nabla\tilde{\nu}\|^2 + \tilde{\nu}\Delta\tilde{\nu}$ and considering φ_i the P_1 Finite Element basis, we rewrite

$$\begin{aligned}\int_{C_i} \frac{\rho}{\sigma}\left(\mathrm{div}\left((\nu+\tilde{\nu})\nabla\tilde{\nu}\right) + c_{b2}\|\nabla\tilde{\nu}\|^2\right)\,\mathbf{dx} &\approx \int_{C_i} \mathrm{div}\left(\frac{\rho_i}{\sigma}\left((\nu+(1+c_{b2})\tilde{\nu})\nabla\tilde{\nu}\right) + c_{b2}\tilde{\nu}_i\nabla\tilde{\nu}\right)\,\mathbf{dx}\\ &= \sum_{j\in\mathcal{V}(i)} \int_{\partial C_i\cap\partial C_j} \left(\frac{\rho_i}{\sigma}\left((\nu+(1+c_{b2})\tilde{\nu})\nabla\tilde{\nu}\right) + c_{b2}\tilde{\nu}_i\nabla\tilde{\nu}\right)\cdot\mathbf{n}_i\,\mathrm{d}\sigma\\ &= -\sum_{K_j\in T(i)} \int_{K_j} \left(\frac{\rho_i}{\sigma}\left((\nu+(1+c_{b2})\tilde{\nu})\nabla\tilde{\nu}\right) + c_{b2}\tilde{\nu}_i\nabla\tilde{\nu}\right)\Big|_{K_j}\cdot\nabla\varphi_i\,\mathbf{dx}.\end{aligned}$$

For the advection term we consider the following approximation

$$\int_{\partial C_i\cap\partial C_j} (\rho\mathbf{v}\tilde{\nu})\cdot\mathbf{n}_i\,\mathrm{d}\sigma \approx \Phi^{\rho}_{ij}(W_i, W_j, \mathbf{n}_{ij})\cdot\begin{cases}\tilde{\nu}_i & \text{if } \Phi^{\rho}_{ij}(W_i, W_j, \mathbf{n}_{ij}) > 0,\\ \tilde{\nu}_j & \text{otherwise,}\end{cases}$$

where Φ^{ρ}_{ij} is the density flux computed with the Roe or HLLC solver. Now considering

$$\tilde{\nu}|_K = \frac{1}{4}\sum_{\ell\in K}\tilde{\nu}_\ell\,,\quad \nabla\tilde{\nu}|_K = \sum_{\ell\in K}\tilde{\nu}_\ell\nabla\varphi_\ell\,,$$

we write the approximation of the diffusion and dissipation terms as follows

$$\begin{aligned}&\int_{K_j}\left(\frac{\rho_i}{\sigma}\left((\nu+(1+c_{b2})\tilde{\nu})\nabla\tilde{\nu}\right) + c_{b2}\tilde{\nu}_i\nabla\tilde{\nu}\right)\Big|_{K_j}\cdot\nabla\varphi_i\,\mathbf{dx}\approx\\ &\qquad\approx |K_j|\frac{\rho_i}{\sigma}\left(\left((\nu|_{K_j} + (1+c_{b2})\tilde{\nu}|_{K_j})\nabla\tilde{\nu}|_{K_j}\right) + c_{b2}\tilde{\nu}_i\nabla\tilde{\nu}|_{K_j}\right)\cdot\nabla\varphi_i|_{K_j}\,.\end{aligned}$$

For source terms we consider the simple following discretization

$$\int_{C_i} f^s\,\mathbf{dx} \approx |C_i|\rho_i\left(c_{b1}\tilde{S}_i\tilde{\nu}_i - c_{w1}f_w\left(\frac{\tilde{\nu}_i}{d_i}\right)^2\right).$$

3.2 Large Eddy Simulation (LES)

The principle of LES is based on a different treatment of the flow structures depending on their size. Large scales are directly solved and the small scales are modeled by what we call a subgrid-scale model. This decomposition is obtained using a filter on the Navier-Stokes equations. Following this filtering operation, fluctuations of a smaller size than the filter sizes are eliminated.

Let φ be a quantity in spectral space, the filtering operation applied to φ is obtained by convolution on the computational domain :

$$\overline{\varphi(\boldsymbol{x},t)} = \int_\Omega \varphi(\boldsymbol{\zeta},t)\,G_\Delta(\boldsymbol{x}-\boldsymbol{\zeta})\,d\boldsymbol{\zeta} = \left(G_\Delta *_{\boldsymbol{x}} \varphi\right)(\boldsymbol{x},t)\,, \tag{3.2}$$

where G is a spatial filter that determines the set of fluctuations representing large scales. The quantity Δ corresponds to the width of the chosen filter, it can be defined by : $\Delta = (\Delta_x \Delta_y \Delta_z)^{1/3}$. For further manipulation of the Navier-Stokes equations after filter application, we require the following properties :

- Consistency

$$\overline{\varphi(\boldsymbol{x}, t)} = \varphi(\boldsymbol{x}, t) \Longleftrightarrow \int_\Omega G_\Delta(\boldsymbol{\zeta})\, d\boldsymbol{\zeta} = 1\,.$$

- Linearity

$$\overline{\varphi + \psi} = \overline{\varphi} + \overline{\psi}\,.$$

- Commutation with differentiation

$$\overline{\frac{\partial \varphi}{\partial \cdot}} = \frac{\partial \overline{\varphi}}{\partial \cdot}\,.$$

For compressible flow we use the Favre filtering

$$\widehat{\varphi} = \frac{\overline{\rho\varphi}}{\overline{\rho}}\,,$$

and after applying our filtering operation to the compressible Navier-Stokes equations, we obtain the equations of motion of the field major flow structures :

$$\begin{cases} \partial_t \overline{\rho} + \operatorname{div}\left(\overline{\rho}\widehat{\mathbf{u}}\right) = 0, \\ \partial_t\left(\overline{\rho}\widehat{\mathbf{u}}\right) + \operatorname{div}\left(\overline{\rho}\widehat{\mathbf{u}} \otimes \widehat{\mathbf{u}}\right) + \nabla \overline{p} = \operatorname{div} \widehat{\mathcal{T}} + \operatorname{div}\left(\overline{\mathcal{T}} - \widehat{\mathcal{T}}\right) - \operatorname{div}\left(\overline{\rho}\left(\widehat{\mathbf{u} \otimes \mathbf{u}} - \widehat{\mathbf{u}} \otimes \widehat{\mathbf{u}}\right)\right), \\ \partial_t(\overline{\rho}\widehat{E}) + \operatorname{div}\left(\left(\overline{\rho}\widehat{E} + \overline{p}\right)\widehat{\mathbf{u}}\right) = \operatorname{div}\left(\lambda \nabla \widehat{T}\right) + \operatorname{div}\left(\widehat{\mathcal{T}} \cdot \widehat{u}\right) + \operatorname{div}\left(\left(\overline{\rho}\widehat{E} + \overline{p}\right)\widehat{\mathbf{u}} - \overline{\left(\rho E + p\right)\mathbf{u}}\right) \\ \qquad\qquad + \operatorname{div}\left(\lambda\left(\nabla \overline{T} - \nabla \widehat{T}\right)\right) + \operatorname{div}\left(\overline{\mathcal{T} \cdot u} - \widehat{\mathcal{T}} \cdot \widehat{u}\right). \end{cases}$$

Let's start with the terms we are going to neglect. Indeed, the terms $\overline{\mathcal{T}} - \widehat{\mathcal{T}}, \overline{\mathcal{T} \cdot u} - \widehat{\mathcal{T}} \cdot \widehat{u}$ and $\lambda\left(\nabla \overline{T} - \nabla \widehat{T}\right)$ will be neglected [132]. For the subgrid-scale terms relating to heat exchange, we have

$$\left(\overline{\rho}\widehat{E} + \overline{p}\right)\widehat{\mathbf{u}} - \overline{\left(\rho E + p\right)\mathbf{u}} = \lambda_{\text{sgs}} \nabla \widehat{T}, \quad \text{where} \quad \lambda_{\text{sgs}} = C_p \frac{\mu_{\text{sgs}}}{Pr_{\text{sgs}}},$$

but note that this term will also be neglected. The last term to look at is the subgrid-scale Reynolds stress tensor $\mathcal{T}_{\text{sgs}} = \overline{\rho}\left(\widehat{\mathbf{u} \otimes \mathbf{u}} - \widehat{\mathbf{u}} \otimes \widehat{\mathbf{u}}\right)$, this stress tensor is separated into deviatoric and isotropic parts as follows

$$\begin{aligned} \mathcal{T}_{\text{sgs}} &= \left(\mathcal{T}_{\text{sgs}} - \frac{1}{3}\text{tr}\left(\mathcal{T}_{\text{sgs}}\right)\mathbb{I}_3\right) + \frac{1}{3}\text{tr}\left(\mathcal{T}_{\text{sgs}}\right)\mathbb{I}_3 \\ &= \mathcal{D}\left(\mathcal{T}_{\text{sgs}}\right) + \mathcal{I}\left(\mathcal{T}_{\text{sgs}}\right). \end{aligned}$$

The deviatoric part is defined by :

$$\mathcal{D}\left(\mathcal{T}_{\text{sgs}}\right) = -\frac{\mu_{\text{sgs}}}{\mu}\widehat{\mathcal{T}},$$

and according to [42] the isotropic part of the subgrid-scale Reynolds stress tensor is given by

$$\mathcal{I}\left(\mathcal{T}_{\text{sgs}}\right) = -\frac{2}{3}\,\overline{\rho}\,(C_I\Delta)^2|\widehat{S}|^2\mathbb{I}_3,\ \text{where } |\widehat{S}|^2 = 2\widehat{S}_{ij}\widehat{S}_{ij} \text{ and } C_I \text{ is a dimensionless constant.}$$

But this model for the isotropic part of the subgrid-scale Reynolds stress tensor is extremely small compared to the thermodynamic pressure, so we neglect it. Finally, our equations can be written as :

$$\begin{cases} \partial_t\overline{\rho} + \operatorname{div}(\overline{\rho}\widehat{\mathbf{u}}) = 0, \\ \partial_t(\overline{\rho}\widehat{\mathbf{u}}) + \operatorname{div}(\overline{\rho}\widehat{\mathbf{u}}\otimes\widehat{\mathbf{u}}) + \nabla\overline{p} = \operatorname{div}\left(\left(1+\frac{\mu_{\text{sgs}}}{\mu}\right)\widehat{\mathcal{T}}\right), \\ \partial_t(\overline{\rho}\widehat{E}) + \operatorname{div}\left(\left(\overline{\rho}\widehat{E}+\overline{p}\right)\widehat{\mathbf{u}}\right) = \operatorname{div}\left(\lambda\nabla\widehat{T}\right) + \operatorname{div}\left(\widehat{\mathcal{T}}\cdot\widehat{\mathbf{u}}\right). \end{cases}$$

Now we have to model μ_{sgs} using a subgrid-scale model.

3.2.1 Smagorinsky model

This model was introduced by J. Smagorinsky in 1963 [124] and was extended to compressible flows by Lesieur et al. in 1996 [77]. For this model the turbulent viscosity is modelled as follows

$$\mu_{\text{sgs}} = \overline{\rho}\,(C_s\Delta)^2\left|\widehat{S}\right|.$$

The term Δ represents the local width of the grid, which we define as $\Delta_k = |T_k|^{\frac{1}{3}}$ where $|T_k|$ represents the measurement of a mesh element T_k. The constant C_s is Smagorinsky's constant and is set at 0.1 .

One of the problems with the Smagorinsky model is that, by construction, it gives a non-zero value for the turbulent viscosity close to the wall. We will present another model that has been introduced with the idea of improving the behaviour of the model in near-wall regions.

3.2.2 WALE model

The WALE (Wall-Adapting Locale Eddy-Viscosity) model was proposed by Nicoud and Ducros [98] in 1999. The turbulent viscosity of this model is expressed in terms of the invariants of the velocity gradient as follows :

$$\mu_{\text{sgs}} = \overline{\rho}\,(C_w\Delta)^2\frac{\left(S^d_{ij}S^d_{ij}\right)^{\frac{3}{2}}}{\left(\widehat{S}_{ij}\widehat{S}_{ij}\right)^{\frac{5}{2}} + \left(S^d_{ij}S^d_{ij}\right)^{\frac{5}{4}}},$$

with

$$S_{ij}^d = \frac{1}{2}\left(\widehat{g}_{ij}^2 + \widehat{g}_{ji}^2\right) - \frac{1}{3}\delta_{ij}\widehat{g}_{kk}^2 \quad \text{where} \quad \widehat{g}_{ij}^2 = \sum_{k=1}^{3} \frac{\partial \widehat{u}_i}{\partial x_k}\frac{\partial \widehat{u}_k}{\partial x_j}.$$

The definition of the filter width Δ is the same as for the Smagorinsky model, and the constant C_w is set to 0.5.

3.2.3 Dynamic model

We have previously seen two subgrid models, namely the Smagorinsky model and the WALE model, where two constants C_s and C_w have been set. In [51] Germano assures us that it is not possible to model the variety of phenomena through a single constant. He therefore proposes a dynamic subgrid model with the aim of adjusting this constant, which will therefore depend on space and time.

The method consists in considering a second filter, a test filter $\widetilde{\cdot}$, with a width greater than that of the grid filter. This filter is applied to the Navier-Stokes equations already filtered by the grid filter, and the additional terms obtained after these filtering steps are then modeled by the subgrid model. The sub-grid term that appears after applying the test filter is

$$\mathcal{T}_{\text{sgs}}^{\text{test}} = \widetilde{\overline{\rho}}\left(\widetilde{\widehat{\mathbf{u}\otimes\mathbf{u}}} - \widetilde{\widehat{\mathbf{u}}}\otimes\widetilde{\widehat{\mathbf{u}}}\right).$$

Recall that the subgrid-scale Reynolds stress tensor, to which our test filter has been applied, is given by

$$\widetilde{\mathcal{T}}_{\text{sgs}} = \widetilde{\overline{\rho}}\left(\widetilde{\widehat{\mathbf{u}\otimes\mathbf{u}}} - \widetilde{\widehat{\mathbf{u}}\otimes\widehat{\mathbf{u}}}\right),$$

and we can now introduce the resolved turbulence tensor $\mathcal{L}$, also known as Leonard's tensor, as follows

$$\mathcal{L} = \mathcal{T}_{\text{sgs}}^{\text{test}} - \widetilde{\mathcal{T}}_{\text{sgs}} = \widetilde{\overline{\rho}}\left(\widetilde{\widehat{\mathbf{u}}\otimes\widehat{\mathbf{u}}} - \widetilde{\widehat{\mathbf{u}}}\otimes\widetilde{\widehat{\mathbf{u}}}\right).$$

It represents the contribution to the Reynolds tensors of the flow scales whose size is between the grid filter and the test filter. The deviatoric part of $\mathcal{L}$ can then be expressed using the Smagorinsky model

$$\begin{aligned}
\operatorname{dev}\mathcal{L} &= \mathcal{L} - \frac{1}{3}\text{tr}\left(\mathcal{L}\right)\mathbb{I}_3 \\
&= \left(\mathcal{T}_{\text{sgs}}^{\text{test}} - \frac{1}{3}\text{tr}\left(\mathcal{T}_{\text{sgs}}^{\text{test}}\right)\mathbb{I}_3\right) - \left(\widetilde{\mathcal{T}}_{\text{sgs}} - \frac{1}{3}\text{tr}\left(\widetilde{\mathcal{T}}_{\text{sgs}}\right)\mathbb{I}_3\right) \\
&= C\Delta^2 \underbrace{\left[\widetilde{\overline{\rho}|\widehat{S}|\widehat{\mathcal{T}}} - \left(\frac{\widetilde{\Delta}}{\Delta}\right)^2 \widetilde{\overline{\rho}}\widetilde{|\widehat{S}|}\widetilde{\widehat{\mathcal{T}}}\right]}_{\mathcal{B}}.
\end{aligned}$$

The problem of determining the constant C for an instantaneous state of flow can be reduced to the following problem $\mathcal{L} - (C\Delta)^2\mathcal{B} = 0$ which by least squares gives

$$(C\Delta)^2 = \frac{\mathcal{B} : \mathcal{L}}{\mathcal{B} : \mathcal{B}}\,.$$

3.3 Detached Eddy Simulation (DES)

We have presented two approaches to modeling turbulence. The first is low-cost (inexpensive), but averages all the fluctuations in the flow, which may make it inaccurate or unsuitable for certain types of study. The second, on the other hand, allows us to capture very interesting turbulence scales, but becomes really expensive as soon as the boundary layer becomes turbulent. So, in this section, we will focus on what we will call a "hybrid model". The idea is to combine the positive aspects of the other two approaches presented above, i.e. to have a model that will be low-cost (inexpensive) in the boundary layer (compared to LES) but more accurate than a RANS model.

In this thesis, we will restrict ourselves to the Detached Eddy Simulation (DES) model, which was originally proposed by [2]. This model is based on the Spalart-Allmaras formulation but uses, instead of wall distance d, a new length scale of which is expressed as a function of grid size Δ as follows

$$\widetilde{d} = \min\left(d,\, C_{DES}\Delta\right)$$

where C_{DES} is a constant of 0.65 and in the case of unstructured meshes, the length Δ is equal to the cell diameter multiplied by $3^{-\frac{1}{2}}$. Close to the wall, $d < C_{DES}\Delta$ the flow is solved via a RANS model and away from the wall, $d > C_{DES}\Delta$ the flow is solved like a LES model. The main disadvantage of the DES method is that the transition between the RANS and LES zones is highly dependent on the local resolution of the mesh, and creates the presence of gray areas when moving from RANS to LES modeling, where turbulence is not physical.

So we shall be looking at a variant of the DES, the Delayed Detached Eddy Simulation (DDES) proposed by [126], which consists in adding a shielding function in order to force the resolution of the boundary layer in RANS, regardless of mesh size. This shielding function f_d is given by :

$$f_d = 1 - \tanh\Big((8\,r_d)^3\Big) \quad \text{where} \quad r_d = \frac{\nu_t + \nu}{\sqrt{\nabla\mathbf{u} : \nabla\mathbf{u}}\,\kappa^2 d^2} \quad \text{with} \quad \kappa = 0.41\,.$$

So the length scale of the Spalart-Allmaras model is redefined as follows :

$$\widetilde{d} = d - f_d\,\max\Big(0; (d - C_{DES}\Delta)\Big).$$

This last version will be used in this thesis.

3.4 Variational Multiscale (VMS) Method

This new approach based on a variational multiscale (VMS) framework was introduced by Hughes and al. [60] (in incompresible), and adapted to compressible flows by [73]. This chapter is based on this last article. Let us start with a reminder, the semi-discretization of the Navier-Stokes system by mixed FV/FE method leads to a system of the form :

$$\int_{C_i} \partial_t W \, \mathbf{dx} + \int_{\partial C_i} F^C(W) \cdot \mathbf{n}_i \, \mathrm{d}\sigma = \int_{\text{supp}\,(\varphi_i)} F^D(W) \cdot \nabla \varphi_i \, \mathbf{dx} \,,$$

which, for practical reasons, is rewritten as follows

$$\int_{\Omega} \partial_t W \, \chi_i \, \mathbf{dx} + \int_{\partial\,\text{supp}\,(\chi_i)} F^C(W) \cdot \mathbf{n}_i \, \chi_i \, \mathrm{d}\sigma = \int_{\text{supp}\,(\varphi_i)} F^D(W) \cdot \nabla \varphi_i \, \mathbf{dx} \,,$$

with χ_i the characteristic function associated with the C_i cell and φ_i the finite element basis functions used to solve the diffusive fluxes. This formulation is also recalled in non-condensed form :

$$\begin{cases} \displaystyle\int_{\Omega} \partial_t \rho \, \chi_i \, \mathbf{dx} + \int_{\partial\,\text{supp}\,(\chi_i)} \rho \mathbf{u} \cdot \mathbf{n}_i \, \chi_i \, \mathrm{d}\sigma = 0, \\ \displaystyle\int_{\Omega} \partial_t (\rho \mathbf{u}) \, \chi_i \, \mathbf{dx} + \int_{\partial\,\text{supp}\,(\chi_i)} \rho \mathbf{u} \otimes \mathbf{u} \cdot \mathbf{n}_i \, \chi_i \, \mathrm{d}\sigma + \int_{\partial\,\text{supp}\,(\chi_i)} p \, \mathbf{n}_i \, \chi_i \, \mathrm{d}\sigma + \int_{\text{supp}\,(\varphi_i)} \mathcal{T} \cdot \nabla \varphi_i \, \mathbf{dx} = 0, \\ \displaystyle\int_{\Omega} \partial_t \left(\rho E\right) \, \chi_i \, \mathbf{dx} + \int_{\partial\,\text{supp}\,(\chi_i)} (\rho E + p) \mathbf{u} \cdot \mathbf{n}_i \, \chi_i \, \mathrm{d}\sigma + \int_{\text{supp}\,(\varphi_i)} \lambda \nabla T \cdot \nabla \varphi_i \, \mathbf{dx} + \int_{\text{supp}\,(\varphi_i)} (\mathcal{T} \cdot \mathbf{u}) \cdot \nabla \varphi_i \, \mathbf{dx} = 0. \end{cases}$$

As with LES, this approach involves separating the flow scales, but this time into three parts : large resolved scales, small resolved scales and unresolved scales. The flow variable is then decomposed as follows :

$$W = \overline{W} + W' + \widehat{W}$$

where $\overline{W}$ denotes the component associated with resolved large scales, W' with resolved small scales and $\widehat{W}$ with unresolved subgrid scales. By denoting by $\mathcal{V}_{FV}$ and $\mathcal{V}_{FE}$ the functional spaces generated respectively by the functions χ_i and φ_i, consider the a priori decomposition

$$\begin{aligned} \mathcal{V}_{FV} &= \overline{\mathcal{V}}_{FV} \oplus \mathcal{V}'_{FV} \oplus \widehat{\mathcal{V}}_{FV} \\ \mathcal{V}_{FE} &= \overline{\mathcal{V}}_{FE} \oplus \mathcal{V}'_{FE} \oplus \widehat{\mathcal{V}}_{FE} \end{aligned}$$

We can rewrite our system governing the large scales resolved using the functions $\overline{\chi_i}$, $\overline{\varphi_i}$ and the decomposition of our variables. The same applies to small scales, which are resolved using the functions χ'_i, φ'_i and the decomposition of our variables. The development of such systems is long and of no great importance for what follows, we'll confine ourselves to its final general form. Please refer to [73] for a detailed decomposition.

As unresolved scales are not captured by numerical resolution, the component noted with " $\hat{.}$ " in the above relations is no longer taken into account (their effects on large resolved scales are neglected while those on small resolved scales are modeled) and the three-level decomposition is therefore reinterpreted as a two-level decomposition :

$$\mathcal{V}_{FV_h} = \overline{\mathcal{V}}_{FV_h} \oplus \mathcal{V}'_{FV_h}$$
$$\mathcal{V}_{FE_h} = \overline{\mathcal{V}}_{FE_h} \oplus \mathcal{V}'_{FE_h}$$

with the following decomposition of our variables

$$W = \overline{W}_h + W'_h$$

where the subscript h denotes the resolved components. The a priori separation of scales is achieved through an agglomeration process. The $\overline{W}_h$ component associated with resolved large scales is then defined using an operator defined in the space of resolved large scales as follows :

$$\overline{W}_h = P(W_h) = \sum_k \overline{\chi}_k W_k = \sum_k \chi_k \widetilde{W}_k,$$

where

$$\overline{\chi}_k = \frac{|C_k|}{\sum\limits_{j\in I_k} |C_j|} \sum_{j\in I_k} \chi_j \quad \text{and} \quad \widetilde{W}_k = \frac{\sum\limits_{j\in I_k} |C_j|\, W_j}{\sum\limits_{j\in I_k} |C_j|},$$

for convective terms, and :

$$\overline{W}_h = P(W_h) = \sum_k \overline{\varphi}_k W_k = \sum_k \varphi_k \widetilde{W}_k,$$

with

$$\overline{\chi}_k = \frac{|C_k|}{\sum\limits_{j\in I_k} |C_j|} \sum_{j\in I_k} \varphi_j,$$

for diffusive terms. We set $I_k = \left\{ j \mid C_j \subset C_{m(k)} \right\}$ where $C_{m(k)}$ is the macro-cell containing C_k.

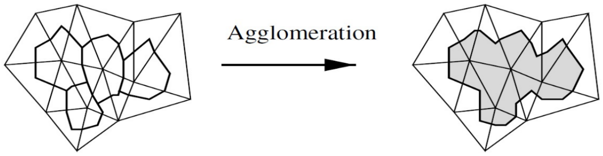


FIGURE 3.1 – An illustration of the agglomeration process.

The resolved small-scale components W_h' are then calculated for the convective terms as follows :

$$\begin{aligned} W_h' &= W_h - \overline{W}_h \\ &= \sum_k \left(\chi_k - \overline{\chi}_k\right) W_k \\ &= \sum_k \chi_k' W_k \\ &= \sum_k \chi_k \left(W_k - \widetilde{W}_k\right), \end{aligned}$$

and for diffusive terms

$$\begin{aligned} W_h' &= W_h - \overline{W}_h \\ &= \sum_k \left(\varphi_k - \overline{\varphi}_k\right) W_k \\ &= \sum_k \varphi_k' W_k \\ &= \sum_k \varphi_k \left(W_k - \widetilde{W}_k\right). \end{aligned}$$

An important process characterizing turbulent flow is the energy cascade, where kinetic energy is transferred from larger to smaller structures. It can be assumed that energy transfer occurs mainly between neighboring scales, from large resolved scales to small resolved scales, and from small resolved scales to unresolved subgrid scales. As a result, another feature of the VMS approach is that the effects of unresolved scales are taken into account only in the equations governing resolved small scales, and are neglected at the level of large structures. These effects are modeled by introducing a turbulence viscosity into the equations associated with the resolved small scales. The following term is then added to the momentum balance equation associated with the resolved small scales :

$$\int_{\text{supp}\,(\varphi_i)} \tau_h' \, \nabla \varphi_i' \, \mathbf{dx},$$

where τ_h' is the subgrid tensor given by :

$$\tau_h' = \mu_{sgs}' \left(2S_{ij}' - \frac{2}{3} S_{kk}' \delta_{ij}\right),$$

with μ_{sgs} the turbulence viscosity, which depends on the subgrid model used and is expressed as a function of the resolved small-scale component of the solution, and

$$S_{ij}' = \frac{1}{2}\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right).$$

Also, the effects of unresolved scales are modeled in the resolved small-scale energy equation by the term :

$$\int_{\text{supp}\,(\varphi_i)} \frac{C_p \mu'_{sgs}}{Pr_{sgs}} \nabla T'_h \cdot \nabla \varphi'_i \,\mathbf{dx}.$$

After bringing together the equations associated with the resolved small scales and those corresponding to the resolved large scales, we obtain the final semi-discretized VMS-LES equations, which are given by the following system (omitting the subscript h) :

$$\begin{cases} \displaystyle\int_\Omega \partial_t \rho \,\mathbf{dx} + \int_{\partial C_i} \rho \mathbf{u} \cdot \mathbf{n}_i \,\mathrm{d}\sigma = 0, \\ \displaystyle\int_\Omega \partial_t (\rho \mathbf{u}) \,\mathbf{dx} + \int_{\partial C_i} \rho \mathbf{u} \otimes \mathbf{u} \cdot \mathbf{n}_i \,\mathrm{d}\sigma + \int_{\partial C_i} p \,\mathbf{n}_i \,\mathrm{d}\sigma + \int_{\text{supp}\,(\varphi_i)} \mathcal{T} \cdot \nabla \varphi_i \,\mathbf{dx} + \int_{\text{supp}\,(\varphi_i)} \tau' \,\nabla \varphi'_i \,\mathbf{dx} = 0, \\ \displaystyle\int_\Omega \partial_t \left(\rho E\right) \,\mathbf{dx} + \int_{\partial C_i} (\rho E + p) \mathbf{u} \cdot \mathbf{n}_i \,\mathrm{d}\sigma + \int_{\text{supp}\,(\varphi_i)} \lambda \nabla T \cdot \nabla \varphi_i \,\mathbf{dx} + \int_{\text{supp}\,(\varphi_i)} (\mathcal{T} \cdot \mathbf{u}) \cdot \nabla \varphi_i \,\mathbf{dx} \\ \displaystyle\qquad\qquad\qquad\qquad + \int_{\text{supp}\,(\varphi_i)} \frac{C_p \mu'_{sgs}}{Pr_{sgs}} \nabla T' \cdot \nabla \varphi'_i \,\mathbf{dx} = 0. \end{cases}$$

CHAPITRE 4

Mesh adaptation

In this thesis, we will focus on the multiscale mesh adaptation for PDE, and more precisely, in the feature-based approach, where we choose a special field, the sensor, which is generally identical to the solution when it is a scalar field. We refer to the monograph [39]. This chapter is devoted to the description of the mesh adaptation methods which will be used in our work. Consider Ω an open of $\mathbb{R}^d$ and $u : \Omega \longrightarrow \mathbb{R}^m$, $m \geqslant 1$, the unknown of our PDE

$$\forall \mathbf{x} \in \Omega, \;\; F\left(\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x}), \dots, \nabla^k u(\mathbf{x})\right) = 0,$$

in which F takes its values in $\mathbb{R}^d$. Consider $\mathcal{H}$, a mesh obtained by discretizing the domain Ω and u_h the numerical solution obtained for our PDE on the mesh $\mathcal{H}$. In the context of mesh adaptation, we would like to control the approximation error $\|u - u_h\|$, for a given norm, in order to generate a new mesh that minimizes this error under a certain constraint. Unfortunately, we do not know how to estimate this error directly. Let V be a Hilbert space and V_h a closed subspace of V and $\Pi_h : V \to V_h$ the linear interpolation operator of u on $\mathcal{H}$. Using Cea's lemma, which applies to elliptic problems, we know that

$$\|u - u_h\|_V \leqslant c\,\|u - \Pi_h u\|_V.$$

We assume that this relation still holds for our class of problems, namely hyperbolic/parabolic, and for non-Hilbertian spaces. Some analysis based on the interpolation error (e.g. Sec. 5.3. of [39]) shows that the link between the interpolation error and the approximation error is stronger than the bound given by Cea's lemma.

1 Continuous Mesh Model

The considered problem of mesh adaptation consist in finding the mesh $\mathcal{H}$ of Ω that minimizes the linear interpolation error $u - \Pi_h u$ in the L^p-norm, for a given sensor u and a given number of mesh vertices N, which we will note $C(\mathcal{H}) = N$:

$$Find \;\; \mathcal{H}_{\text{opt}} = \operatorname{Arg}\min_{\mathcal{H}} \|u - \Pi_h u\|_{L^p(\Omega_h)} \;\; such\ that \;\; C(\mathcal{H}) = N.$$

This error model highly depends on the topology of the mesh which turns out to be intractable practically. Therefore, we recast it in the continuous mesh framework that we will introduce here.

1.1 Metric space concepts

Definition 1.1 Metric space. *We call a Euclidean metric space a vector space with a scalar product* $\langle\cdot,\cdot\rangle_{\mathcal{M}}$ *defined by a metric tensor, i.e. a symmetric positive definite matrix* $\mathcal{M}$*. It is noted* $(\mathbb{R}^d,\mathcal{M})$*. The scalar product of two vectors* $\mathbf{u}$ *and* $\mathbf{v}$ *is given by* $\langle\mathbf{u},\mathbf{v}\rangle_{\mathcal{M}} = {}^t\mathbf{u}\,\mathcal{M}\,\mathbf{v}$.

From a Euclidean metric space $(\mathbb{R}^d,\mathcal{M})$ we can then define a normed vector space $(\mathbb{R}^d,\|\cdot\|_{\mathcal{M}})$ whose structure is induced by $\mathcal{M}$. We note $\|\mathbf{u}\|_{\mathcal{M}} = \sqrt{{}^t\mathbf{u}\,\mathcal{M}\,\mathbf{u}}$.

Let $\Omega \subset \mathbb{R}^d$ a bounded convex domain such that $\partial\Omega$ is polygonal. Consider the conformal discretization $\mathcal{H}$ of Ω, let K be the elements of $\mathcal{H}$ and $\mathbf{e}_K$ the edges of the element K. Consider the set S_d of symmetric $d\times d$ positive definite matrices. For each non-degenerate element there exists a matrix (unique if $d=2$) $\mathcal{M}\in S_d$ such that, for a given constant $C>0$

$$\|\mathbf{e}_K^i\|^2_{\mathcal{M}} = {}^t\mathbf{e}_K^i\,\mathcal{M}\,\mathbf{e}_K^i = C,\ \ \forall i\in\{1,...,6\}\text{ if } d=3,\ \forall i\in\{1,2,3\}\text{ if } d=2. \tag{4.1}$$

The tetrahedron is thus regular or the triangle is thus equilateral in the norm induced by the metric $\mathcal{M}$ and the mesh element is said to be unit with respect to the local this metric. A unit element is inscribed in the ellipse $E := \{\mathbf{x}\in\mathbb{R}^d;\ {}^t\mathbf{x}\,\mathcal{M}\,\mathbf{x}\leqslant 1\}$ with respect to $\mathcal{M}$ and a prescribed constant C in (4.1).

- $2D$: Since $\mathcal{M}$ is a positive definite symmetric matrix, we have the following unique decomposition

$$\mathcal{M} = \mathcal{R}\begin{pmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{pmatrix}{}^t\mathcal{R},\quad \text{with}\ \ \mathcal{R} = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}.$$

The eigenvectors then give the directions of the principal axes, and the eigenvalues give $\lambda_1 = h_1^{-2},\ \lambda_2 = h_2^{-2}$, where h_i are the sizes prescribed by $\mathcal{M}$. We also define the local density $d=\sqrt{\lambda_1\lambda_2} = (h_1h_2)^{-1}$. Denoting by $r_i = h_i^2\,(h_1h_2)^{-1}$ the anisotropic quotients, we rewrite the continuous mesh locally

$$\mathcal{M}(\mathbf{x}) = d(\mathbf{x})\,\mathcal{R}(\mathbf{x})\begin{pmatrix}r_1^{-1}(\mathbf{x}) & 0\\ 0 & r_2^{-1}(\mathbf{x})\end{pmatrix}{}^t\mathcal{R}(\mathbf{x}).$$

- $3D$: Since $\mathcal{M}$ is a positive definite symmetric matrix, we have the following decomposition

$$\mathcal{M} = \mathcal{R}\begin{pmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{pmatrix}{}^t\mathcal{R}.$$

The eigenvectors then give the directions of the principal axes, and the eigenvalues give $\lambda_1 = h_1^{-2},\ \lambda_2 = h_2^{-2}$ and $\lambda_3 = h_3^{-2}$, where h_i are the sizes prescribed by $\mathcal{M}$. We also define the local density $d = \sqrt{\lambda_1\lambda_2\lambda_3} = (h_1h_2h_3)^{-1}$. Denoting by $r_i = h_i^3\,(h_1h_2h_3)^{-1}$ the anisotropic quotients, we rewrite the continuous mesh locally

$$\mathcal{M}(\mathbf{x}) = d^{\frac{2}{3}}(\mathbf{x})\,\mathcal{R}(\mathbf{x}) \begin{pmatrix} r_1^{-\frac{2}{3}}(\mathbf{x}) & 0 & 0 \\ 0 & r_2^{-\frac{2}{3}}(\mathbf{x}) & 0 \\ 0 & 0 & r_3^{-\frac{2}{3}}(\mathbf{x}) \end{pmatrix} {}^t\mathcal{R}(\mathbf{x}).$$

1.2 Riemann metric

We now turn to the more general case where the metric varies from one point in the domain $\Omega \subset \mathbb{R}^d$ to another. We have the following definition (see Sec. 5.4. and in particular Definition 3.4.1. of [39])

Definition 1.2 *Let* $\mathcal{M} : \Omega \to S_d$ *be an integrable map. In addition, let* $\mathbf{u}_0, \mathbf{u}_1 \in \mathbb{R}^d$ *such that* $\mathbf{u}_0 \in \Omega$ *and* $\mathbf{u}_0 + \mathbf{u}_1 \in \Omega$. *Consider the parametrization* $\mathbf{u} : [0,1] \to \mathbb{R}^d,\ \mathbf{u}(t) = \mathbf{u}_0 + t\mathbf{u}_1,\ t \in [0,1]$, *and we set*

$$\|\mathbf{u}\|_{\mathcal{M}} := \int_0^1 \sqrt{{}^t\mathbf{u}'(t)\mathcal{M}(\mathbf{u}_0 + t\mathbf{u}_1)\mathbf{u}'(t)}\,\mathrm{d}t = \int_0^1 \sqrt{{}^t\mathbf{u}_1\mathcal{M}(\mathbf{u}_0 + t\mathbf{u}_1)\mathbf{u}_1}\,\mathrm{d}t.$$

We call $\mathcal{M}$ *the Riemann metric on* Ω *and* $\|\mathbf{u}\|_{\mathcal{M}}$ *the size of edge* $\mathbf{u}$ *in the Riemann metric* $\mathcal{M}$.

Remark 1.1 *If the metric* $\mathcal{M}$ *is constant along* $\mathbf{u}$ *then we have* $\|\mathbf{u}\|_{\mathcal{M}} = \sqrt{{}^t\mathbf{u}_1\,\mathcal{M}\,\mathbf{u}_1}$.

Definition 1.3 *Given a bounded subset* $K \subset \Omega$, *the volume of* K *computed with respect to Riemannian metric space* $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ *is :*

$$|K|_{\mathcal{M}} = \int_K \sqrt{\det \mathcal{M}(\mathbf{x})}\,\mathrm{d}\mathbf{x}.$$

The advantage of having a space in which the scalar product is not the same at every position in space is that you have a different unit ball everywhere, unlike a Euclidean space, where the unit ball is the same everywhere. So we use a Riemannian metric space to compute the necessary geometrical quantities to generate a unit mesh.

A unit mesh with respect to the Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is a mesh whose elements are all unity for this metric. An element K, defined by its list of edges $\{\mathbf{e}_i\}$, is said to be a unit element with respect to the metric $\mathcal{M}$ if the length of all its edges is unit :

$$\|\mathbf{e}_i\|_{\mathcal{M}} = \sqrt{{}^t\mathbf{e}_i\,\mathcal{M}\,\mathbf{e}_i} = 1.$$

2 Interpolation error optimisation

The question now is, what is our model error in the continuous mesh framework. According to [82], [83], if we denote by $\pi_{\mathcal{M}}$ the continuous linear interpolate and H_u the Hessian of the sensor u, then the following continuous linear interpolation estimate holds

$$\forall \mathbf{x} \in \Omega, \ \ |u - \pi_{\mathcal{M}} u| (\mathbf{x}) = K_x \, \mathrm{trace} \left(\mathcal{M}(\mathbf{x})^{-\frac{1}{2}} \, |H_u(\mathbf{x})| \, \mathcal{M}(\mathbf{x})^{-\frac{1}{2}} \right),$$

with $K_x = \frac{1}{8}$ in $2D$ and $K_x = \frac{1}{20}$ in $3D$. Moreover, if $\mathcal{H}$ is a unit mesh with respect to $\mathcal{M}$ and u a smooth function, then we have the following inequality

$$\| u - \Pi_h u \|_{L^p(\Omega_h)} \lesssim \| u - \pi_{\mathcal{M}} u \|_{L^p(\Omega)} .$$

So in the continuous mesh framework the error model is now given by

$$\mathcal{E}(\mathcal{M}) = K_x \left(\int_\Omega \mathrm{trace} \left(\mathcal{M}(\mathbf{x})^{-\frac{1}{2}} \, |H_u(\mathbf{x})| \, \mathcal{M}(\mathbf{x})^{-\frac{1}{2}} \right)^p \mathrm{d}\mathbf{x} \right)^{\frac{1}{p}}, \tag{4.2}$$

and the spatial size of the continuous mesh is given by its spatial complexity :

$$\mathcal{C}(\mathcal{M}) = \int_\Omega \sqrt{\det \mathcal{M}(\mathbf{x})} \, \mathrm{d}\mathbf{x} = \mathcal{N}.$$

Finally the considered problem of mesh adaptation consists in finding the continuous mesh $\mathcal{M}$ that minimizes the continuous linear interpolation error $u - \pi_{\mathcal{M}} u$ in the L^p-norm, for a given sensor u and a given spatial complexity $\mathcal{C}(\mathcal{M}) = \mathcal{N}$:

$$Find \ \ \mathcal{M}_{\mathrm{opt}} = \mathrm{Arg}\min_{\mathcal{M}} \| u - \pi_{\mathcal{M}} u \|_{L^p(\Omega)} \ \ such \ that \ \ \mathcal{C}(\mathcal{M}) = \mathcal{N}. \tag{4.3}$$

The analytical expression of the optimal continuous mesh $\mathcal{M}_{\mathrm{opt}}$ solution of this problem is [83] :

$$\mathcal{M}_{\mathrm{opt}}(\mathbf{x}) = \mathcal{N}^{\frac{2}{d}} \left(\int_\Omega \det\left(|H_u(\mathbf{x})|\right)^{\frac{p}{2p+d}} \mathrm{d}\mathbf{x} \right)^{-\frac{2}{d}} \det\left(|H_u(\mathbf{x})|\right)^{-\frac{1}{2p+d}} |H_u(\mathbf{x})| , \tag{4.4}$$

and for this optimal metric expression the associated optimal interpolation error is

$$\mathcal{E}(\mathcal{M}_{\mathrm{opt}}) = d \, K_x \, \mathcal{N}^{-\frac{2}{d}} \left(\int_\Omega \det\left(|H_u(\mathbf{x})|\right)^{\frac{p}{2p+d}} \mathrm{d}\mathbf{x} \right)^{\frac{2p+d}{dp}} . \tag{4.5}$$

According to [82], [83], $\mathcal{M}$ verifies the following properties :

- $\mathcal{M}_{\mathrm{opt}}$ is unique

- $\mathcal{M}_{\text{opt}}$ is locally aligned with the eigenvectors basis of H_u and has the same anisotropic quotients as H_u
- For a sequence of continuous meshes having an increasing complexity with the same orientation and anisotropic quotients $(\mathcal{M}_{\text{opt}}^{\mathcal{N}})_{\mathcal{N}}$, the asymptotic order of convergence verifies :

$$\|u - \pi_{\mathcal{M}_{\text{opt}}^{\mathcal{N}}} u\|_{L^p(\Omega)} \leqslant \frac{Cst}{\mathcal{N}^{2/3}}.$$

This relation points out a global second order of mesh convergence.

Remark 2.1 *Note that if we consider an interpolation error of order q, approximated by the pseudo-Hessian formulation (e.g. reconstruction of a pseudo-Hessian by least-squares formula), we have the following result :*

$$\mathcal{M}_{\text{opt}}(\mathbf{x}) = \mathcal{N}^{\frac{2}{d}} \left(\int_{\Omega} \det\left(|H_u(\mathbf{x})|\right)^{\frac{pq}{2(pq+d)}} \mathrm{d}\mathbf{x} \right)^{-\frac{2}{d}} \det\left(|H_u(\mathbf{x})|\right)^{-\frac{1}{pq+d}} |H_u(\mathbf{x})| ,$$

and the associated optimal interpolation error is

$$\mathcal{E}(\mathcal{M}_{\text{opt}}) = d^{\frac{q}{2}} \mathcal{N}^{-\frac{q}{d}} \left(\int_{\Omega} \det\left(|H_u(\mathbf{x})|\right)^{\frac{pq}{2(pq+d)}} \mathrm{d}\mathbf{x} \right)^{\frac{pq+d}{dp}} .$$

2.1 Algorithm

For steady simulations, an adaptive computation is carried out via a mesh adaptation loop inside which an algorithmic (or iterative) convergence of the couple mesh-solution is sought, in the sense that the solution is algorithmically (or iteratively) converging toward the steady state solution and the mesh is converging toward the adapted mesh associated to this converged steady state. In a feature-based approach, the user chooses a particular solution-dependant field, the sensor (or feature). If the solution field is a scalar field, the solution is taken as sensor.

Algorithm 4.1 – Feature-based mesh adaptation, (Sec. 5.6. of [39])

```
Initial mesh $\mathcal{H}_0$, solution $\mathcal{S}_0$ and complexity $\mathcal{N}$.

  while $i \leqslant n_{adap}$ do
    1. Compute optimal metric for the sensor interpolation error estimate and complexity
    $\Rightarrow \mathcal{M}_i$
    2. Generate new adapted mesh $\Rightarrow \mathcal{H}_i$
    3. Interpolation state on the new mesh $\Rightarrow (\mathcal{S}_0)_i$
    4. Compute state $\Rightarrow \mathcal{S}_i$
    if (solution not fully adapted) then
      $i = i + 1$
    else
      Stop
    end if
  end while
```

At each stage, a numerical solution is computed on the current mesh and is analyzed by means of the error estimate. Next, a unit mesh is generated with respect to this metric. Finally, the solution is interpolated on the new adapted mesh. This procedure is repeated until the algorithmic convergence of the solution and of the mesh is achieved.

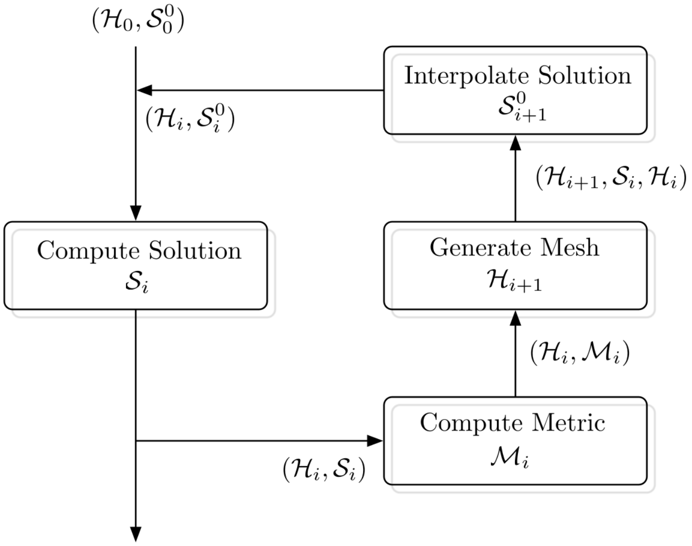


FIGURE 4.1 – Schematic presentation of the feature-based mesh adaptation algorithm. Symbols $\mathcal{H}$, $\mathcal{S}$ and $\mathcal{M}$ holds respectively for mesh, flow solution and metric.

2.2 Numerical examples

As a simple example, let us consider a flow around a triangle at Mach 1.47, in a pipe. This is a stationary RANS simulation and the stationary adaptation algorithm is perfectly suited to capture the solution to this flow (Figure 4.2). We proceed as follows, we carry out an initial calculation on an initial mesh with a fixed number of iterations (500 here), once this number of iterations has been reached (or a convergence condition has been reached) we use the last solution calculated to generate a new mesh. We interpolate our last solution on our new mesh and continue our calculation until we reach the fixed number of iterations, and so on. For the convergence process, the imposed complexity is not fixed. In fact, we impose an increase in complexity at a certain number of remeshing iterations (or a convergence condition), for example we double the complexity every five remeshing iterations, as in our example.

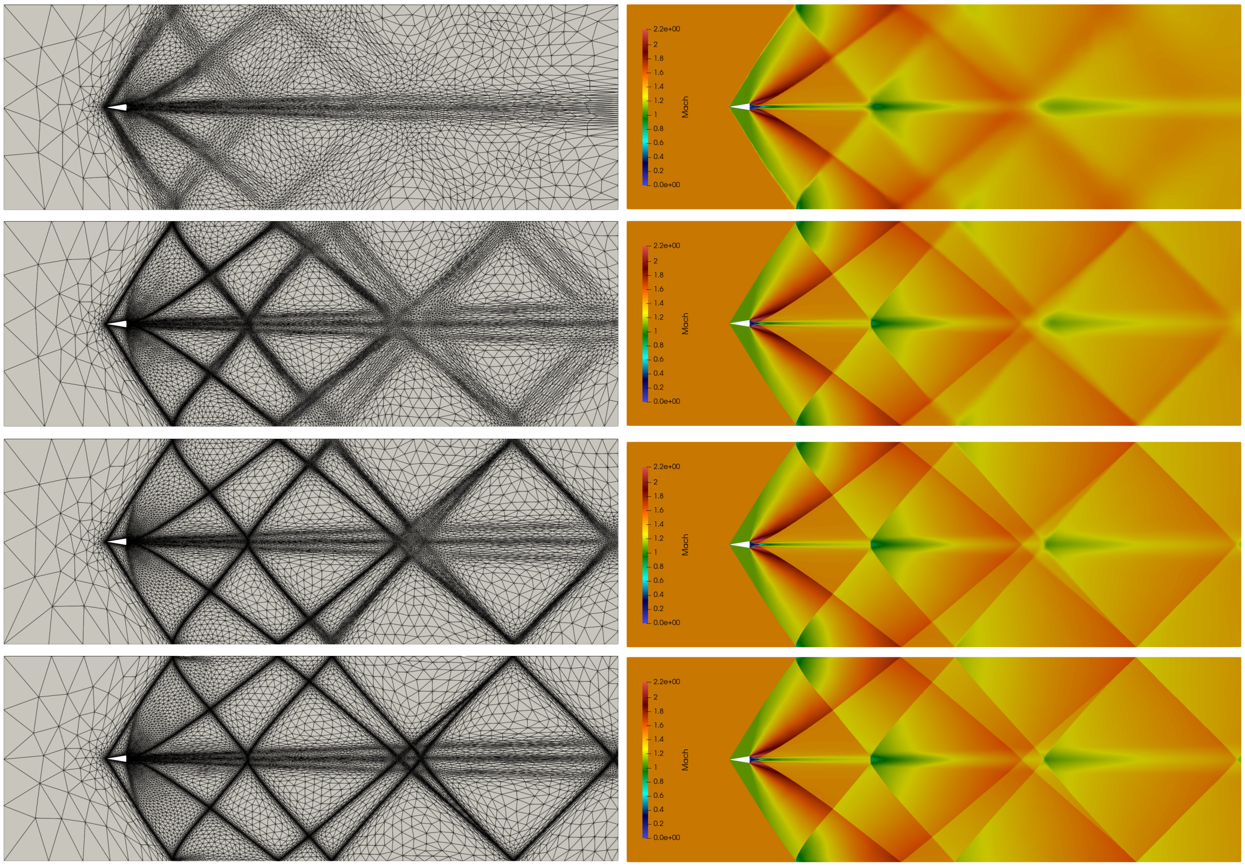


FIGURE 4.2 – Flow around a triangular obstacle in a pipe at Mach 1.47 . View of the mesh and solution for remeshing iterations 2, 5, 8 and 15.

3 Mesh adaptation for unsteady problems

We have already seen the mesh adaptation method for stationary problems, but what about the unsteady case ? Unsteady phenomena such as turbulent phenomena and moving phenomena are more difficult to capture using mesh adaptation, and especially impossible using the adaptation method presented above.

3.1 State of the art

Research into mesh adaptation for stationary problems is rich, which is not the case for unsteady problems.

For example, mention should be made of [134] (1990), which developed an unsteady mesh adaptation algorithm based on very regular error estimation during the calculation. If the error is greater than a prescribed value, we remesh.

We should also mention the mesh refinement methods for unsteady flows used, for example, in [81] and [109] in 1992. This method involves refining/coarsening the mesh very frequently, keeping the solution within the refined zone.

But the story we are interested in begins in 2007 ([10]), with the transient fixed-point algorithm. This is the first works on metric-based adaptation for unsteady flows and it is this theory that we are going to present in this Chapter. Finally, note that there is an extension to the transient fixed-point algorithm, named global fixed-point mesh adaptation algorithm [13]. This algorithm will be mentioned in Chapter 8, where an extension will be given.

3.2 Transient fixed point-based mesh adaptation

Now we solve an unsteady PDE defined on the computational space-time domain $Q = \Omega \times [0,T]$ where T is the maximal time and $\Omega \subset \mathbb{R}^d$ the spatial domain. This time let us imagine that we want to minimize $\|u - \pi_{\mathcal{M}} u\|_{L^p([0,T],L^p(\Omega))}$, still under our constraint $\mathcal{C}(\mathcal{M}) = \mathcal{N}$. We choose to control the worst error, and consider the following inequality, for $1 \leqslant p < \infty$,

$$\|u - \pi_{\mathcal{M}} u\|_{L^p([0,T],L^p(\Omega))} \leqslant T^{\frac{1}{p}} \|u - \pi_{\mathcal{M}} u\|_{L^\infty([0,T],L^p(\Omega))}.$$

The idea is as follows. We divide our time interval into n_{adap} sub-intervals $[t_{i-1}, t_i]$, and we choose to generate a single mesh for each sub-interval which must "take account of the average flow over this sub-interval". In this context, the metric $\mathcal{M}$ is considered to be constant in a sub-interval, i.e. $\mathcal{M}(\mathbf{x}, t) = \mathcal{M}^i(\mathbf{x})$ over the sub-interval $[t_{i-1}, t_i]$. So for this sub-interval we have

$$\begin{aligned}\|u-\pi_{\mathcal{M}}u\|_{L^p([t_{i-1},t_i],L^p(\Omega))} &\leqslant (t_i-t_{i-1})^{\frac{1}{p}}\|u-\pi_{\mathcal{M}}u\|_{L^\infty([t_{i-1},t_i],L^p(\Omega))}\\ &= K_x\left(\int_\Omega \text{trace}\left(\mathcal{M}^i(\mathbf{x})^{-\frac{1}{2}}\,(t_{i-1},t_i)^{\frac{1}{p}}\max_{t\in[t_{i-1},t_i]}|H_u(\mathbf{x},t)|\,\mathcal{M}^i(\mathbf{x})^{-\frac{1}{2}}\right)^p \mathrm{d}\mathbf{x}\right)^{\frac{1}{p}},\end{aligned}$$

and by setting $\mathbf{H}_u^i(\mathbf{x})=(t_i-t_{i-1})^{\frac{1}{p}}\max_{t\in[t_{i-1},t_i]}|H_u(\mathbf{x},t)|$ we have the form of the error we want to control, which is nothing other than our error (4.2) with a different Hessian metric,

$$\|u-\pi_{\mathcal{M}}u\|_{L^p([t_{i-1},t_i],L^p(\Omega))}\leqslant K_x\left(\int_\Omega \text{trace}\left(\mathcal{M}^i(\mathbf{x})^{-\frac{1}{2}}\mathbf{H}_u^i(\mathbf{x})\,\mathcal{M}^i(\mathbf{x})^{-\frac{1}{2}}\right)^p\mathrm{d}\mathbf{x}\right)^{\frac{1}{p}}.$$

For each sub-interval $[t_{i-1},t_i]$, under the spatial complexity $\mathcal{C}(\mathcal{M}^i)=\int_\Omega\sqrt{\det\mathcal{M}^i(\mathbf{x})}\,\mathrm{d}\mathbf{x}=\mathcal{N}^i$, we want to minimize the error

$$\mathcal{E}^i(\mathcal{M}^i)=K_x\left(\int_\Omega \text{trace}\left(\mathcal{M}^i(\mathbf{x})^{-\frac{1}{2}}\mathbf{H}_u^i(\mathbf{x})\,\mathcal{M}^i(\mathbf{x})^{-\frac{1}{2}}\right)^p\mathrm{d}\mathbf{x}\right)^{\frac{1}{p}},$$

and as previously, we get the spatial optimality condition

$$\mathcal{M}^i_{\text{opt}}(\mathbf{x})=(\mathcal{N}^i)^{\frac{2}{d}}\left(\int_\Omega\left(\det\mathbf{H}_u^i(\mathbf{x})\right)^{\frac{p}{2p+d}}\mathrm{d}\mathbf{x}\right)^{-\frac{2}{d}}\left(\det\mathbf{H}_u^i(\mathbf{x})\right)^{-\frac{1}{2p+d}}\mathbf{H}_u^i(\mathbf{x}),$$

with the corresponding optimal error

$$\mathcal{E}^i(\mathcal{M}^i_{\text{opt}})=d\,K_x(\mathcal{N}^i)^{-\frac{2}{d}}\left(\int_\Omega\left(\det\mathbf{H}_u^i(\mathbf{x})\right)^{\frac{p}{2p+d}}\mathrm{d}\mathbf{x}\right)^{\frac{2p+d}{dp}}.$$

3.2.1 Algorithm

To describe what happens in practice, let us start with the basic idea of dividing the time interval $[0,T]$ into n_{adap} sub-intervals

$$[0,T]=\bigcup_{i=1}^{n_{adap}}[t_{i-1},t_i],$$

The solution is calculated over the sub-interval $[t_0,t_1]$, the adaptation variable u is sampled at regular time intervals (over our sub-interval $[t_0,t_1]$), so we have n_k samples of our solution. The Hessian associated with each sample is calculated, so we have n_k Hessians. In practice, we calculate a Hessian from these Hessian matrix fields as follows

$$\mathbf{H}_u^i(\mathbf{x})\approx(t_i-t_{i-1})^{\frac{1}{p}}\bigcap_{k=1}^{n_k}|H_u(\mathbf{x},t^k)|=(t_i-t_{i-1})^{\frac{1}{p}}|H^i_{\text{max}}(\mathbf{x})|,$$

where $\cap$ has to be understood as the metric intersection. We compute $\mathbf{H}_u^1(\mathbf{x})$ in this way for the sub-interval $[t_0, t_1]$ and the resulting metric will prescribe for each node the maximum dimensions guaranteeing control of the spatial interpolation error on this first interval. A new adapted mesh for this sub-interval is generated, our solution at t_0 is interpolated on our obtained mesh and the solution for the interval $[t_0, t_1]$ is recalculated. This is what we call a fixed point iteration, this quantity is prescribed by the user and we will denote it by n_{ptfx}, this procedure is repeated on the subinterval until n_{ptfx} is reached. Once the fixed point iterations have been completed for a sub-interval, we move on to the next one and repeat the same procedure.

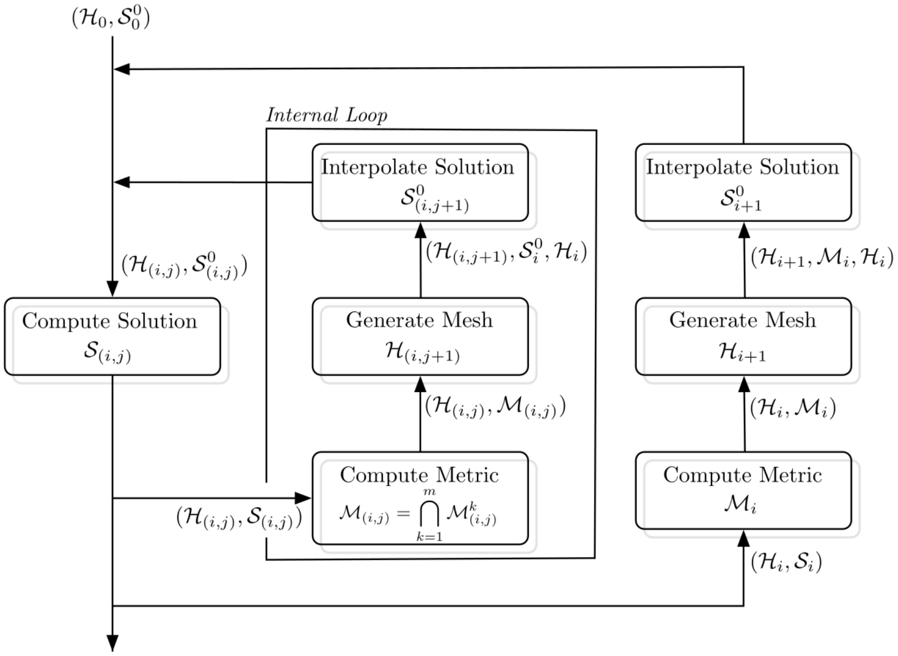


FIGURE 4.3 – Schematic presentation of the transient fixed-point mesh adaptation algorithm. Symbols $\mathcal{H}$, $\mathcal{S}$ and $\mathcal{M}$ hold respectively for mesh, flow solution and metric.

Algorithm 4.2 – Transient $\mathbf{L}^{\infty}\left([0,T],\mathbf{L}^{p}(\Omega)\right)$ fixed-point mesh adaptation algorithm, (e.g. Chap. 8. of [39])

```
//- Loop over time subintervalls $n_{adap}$
for $i = 1, n_{adap}$ do
    //- Fixed point adaptation loop
    for $i = 1, n_{ptfx}$ do
        1. $\mathcal{S}_{0,j}^{i}$ = ConservativeSolutionTransfer($\mathcal{H}_{j}^{i-1}, \mathcal{S}_{j}^{i-1}, \mathcal{H}_{j}^{i}$)
        2. $\mathcal{S}_{j}^{i}$ = SolveStateForward($\mathcal{S}_{0,j}^{i}, \mathcal{H}_{j}^{i}$)
        3. $\mathcal{M}_{j}^{i}$ = ComputeFeatureOrientedMetric($\mathcal{S}_{j}^{i}, \mathcal{H}_{j}^{i}$)
        4. $\mathcal{H}_{j+1}^{i}$ = GenerateAdaptedMesh($\mathcal{H}_{j}^{i}, \mathcal{M}_{j}^{i}$)
    end for
end for
```

CHAPITRE 5

Central Essentially Oscillating (CENO) approximation

1 Introduction

For about a decade, the teams of Norma have developed a reflexion focusing on the use of more accurate approximation schemes in particular for a better resolution of vortices in flows and sound propagation.

High-order methods are a mean for maximizing the efficiency of approximation in PDE-based simulation. For a given mesh, they are more expensive by some factor, let us say η, than second-order approximations. Now η does not change when the mesh fineness is increased. In contrast, the improvement factor in accuracy, tends to infinity with increased mesh fineness. The gain in global efficiency, which can be measured as the necessary CPU cost for a prescribed error level, can be large as far as high-order convergence is really obtained.

Note that a first practical limitation of this property for industrial CFD is the fact that a very small error, typically 0.001% is definitively not necessary for industrial flows for which the modelling error (error between physical solution and non-discretized modeled solution) is of order of 1%. This probably means that, as long as modelling error is not drastically improved, convergence order higher than 5 might not be advantageous. The present work addresses the fourth-order accurate treatment of advection in 3D Navier-Stokes.

While many publications propose Discontinuous Galerkin higher-order approximations for Navier-Stokes [20, 35, 34, 121], our standpoint is to consider a finite volume approximation, the CENO approximation. The CENO approximation has been introduced and then designed in a series of works, [64, 65, 96, 97, 101, 55, 67, 66, 32, 107, 118]. A comparison between higher order methods can be found in [14]. A disadvantage of CENO is the need of a even higher reconstruction order for dealing with viscous terms. An advantage of CENO is a thin capture of discontinuities potentially under the form of a crisp transition between two neighboring degrees of freedom.
In the present paper, our first contribution is to define a vertex-centered cubic CENO formulations with dual cells, extending the approximation in [28], similar to (but different from) the approximations of [118].

The pro and cons for choosing cell-centering or vertex centering are discussed in detail in these references. In the context of this thesis, an advantage of CENO-vertex is

that it is a rather direct extension of MUSCL-vertex and inherits of :
- most MUSCL-FEM data structures including mesh description,
- a large part of numerical algorithms,
- mesh adaptation technologies.

We have explained in the previous chapter that the MUSCL approximation combined cleverly finite-volume conservative integration and finite difference based reconstruction, and that it however leads to limitations in the increasing of accuracy error for MUSCL-FEM on completely unstructured meshes. The vertex CENO formulation of this chapter keeps the basic representation of the unknown, and keeps the finite-volume conservative integration, but recover the coherence necessary for high order by using a reconstruction based on cell means.

An important disadvantage of vertex CENO is the extra computational cost of the vertex-centered option. A second contribution of this chapter is propositions for CPU optimization of the CENO-vertex approximation.

The plan of the paper is as follows :
Section 2 defines polynomial reconstruction and deals with the relation of reconstruction errors with interpolation errors. Section 3 describes the application of the CENO approximation to Navier-Stokes. Section 4 studied several efficiency improvements for this approximation.

2 Interpolation and reconstruction error

Many high-order approximation schemes like Discontinous Galerkin [20, 34, 35, 121], ENO [18, 41, 55, 75] or distributive schemes [5] use kth-order interpolation or reconstruction and are k-exact (we precise this k-exact in the sequel of this work). Most analyses of k-exact reconstructions are inspired by the Bramble-Hilbert principle, saying that an approximation which is exact for kth-order polynomial is a $(k+1)$th-order accurate approximation. Demonstrations can be found in the fundamental paper on interpolation [33]. Later, when considering reconstruction-based schemes, the authors in e.g. [43] refered to the Taylor series. A re-visiting of that question is found in the ENO paper [4] which establishes the link with [33].

2.1 Notations

Let us consider a smooth real multivariable multidimensional function F of several real variables in $\mathbb{R}^d$ and taking values in $\mathbb{R}^r$:

$$F : \mathbb{R}^d \to \mathbb{R}^r \quad ; \quad \mathbf{x} \mapsto F(\mathbf{x}).$$

We give now several notations for the Taylor series (if convergent) for F around a particular value $\mathbf{c}$ of $\mathbb{R}^d$. First in a compact manner :

$$F(\mathbf{x}) = \sum_{m=0}^{\infty} \frac{1}{m!} F^{(m)}_{\mathbf{x},\mathbf{c}}.$$

Introducing $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_d), \;\; |\boldsymbol{\alpha}| = \sum_k \alpha_k$, we have :

$$F^{(m)}_{\mathbf{x},\mathbf{c}} = \sum_{|\boldsymbol{\alpha}|=m} \binom{m}{\boldsymbol{\alpha}} (\mathbf{x}-\mathbf{c})^{\boldsymbol{\alpha}} D^{\boldsymbol{\alpha}} F(\mathbf{c})$$

with :

$$D^{\boldsymbol{\alpha}} = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial x_2^{\alpha_2}} ... \frac{\partial^{\alpha_d}}{\partial x_d^{\alpha_d}} \quad ; \quad \binom{m}{\boldsymbol{\alpha}} = \frac{m!}{\alpha_1!\alpha_2!...\alpha_d!}.$$

We note that for $|\boldsymbol{\alpha}| = m$, we have m factors in $(\mathbf{x}-\mathbf{c})^{\boldsymbol{\alpha}}$:

$$(\mathbf{x}-\mathbf{c})^{\boldsymbol{\alpha}} = (x_{k_1} - c_{k_1})(x_{k_2} - c_{k_2})...(x_{k_m} - c_{k_m})$$

with, for example, when $\boldsymbol{\alpha} = (2, 3, 1)$,

$$k_1 = k_2 = 1, \; k_3 = k_4 = k_5 = 2, \; k_6 = 3,$$

that is :

$$(\mathbf{x}-\mathbf{c})^{\boldsymbol{\alpha}} = (x - c_1)^2 (y - c_2)^3 (z - c_3).$$

Then, putting $\delta x_{k_\ell} = (x_{k_\ell} - c_{k_\ell}), \; \ell = 1, ...m$, the scalar function

$$\delta\mathbf{x} \;\; \mapsto \;\; G(\delta\mathbf{x}) = (\delta\mathbf{x})^{\boldsymbol{\alpha}}$$

is a m-linear form $G(\delta\mathbf{x}_1, ..., \delta\mathbf{x}_m)$ where G is linear with respect to $\delta\mathbf{x}_k$, for any $k = 1, ..., m$. And since F takes its values in $\mathbb{R}^d$, $\frac{1}{m!} F^{(m)}_{\mathbf{x},\mathbf{c}} \cdot (y_1, ..., y_r)$ is the action of $\mathbb{T}^{(m)}_{F,\mathbf{c}}$, a (m, r) tensor[1] on $\Big(\delta x_{k_1}, ...\delta x_{k_m}, y_1, ..., y_r\Big)$:

$$\frac{1}{m!} F^{(m)}_{\mathbf{x},\mathbf{c}} \cdot (y_1, ..., y_r) = \mathbb{T}^{(m)}_{F,\mathbf{c}}(\delta x_{k_1}, ...\delta x_{k_m}, y_1, ..., y_r) \in \mathbb{R}.$$

In the case where $F(\mathbf{x})$ is scalar, i.e. $r = 1$, we shall use for the m-th term of Taylor series the compact notation :

$$\frac{1}{m!} F^{(m)}_{\mathbf{x},\mathbf{c}} = \mathbb{T}^{(m)}_F(\mathbf{c}) \cdot \boldsymbol{\delta}\mathbf{x}.$$

---

[1] Multilinear form on $V \times ... \times V \times V^* \times ... \times V^*$ with m times V and r times V^*.

2.2 Reconstruction error

Let us focus in this section on the estimation of the *reconstruction error*. Given a function sufficiently smooth u defined on a bounded domain Ω of $\mathbb{R}^3$ limited by a continuous boundary, given a tesselation of Ω into cells C_i of centroids $\mathbf{c}_i$, and the array $\bar{u} = \{\bar{u}_i\}$ of means of u on cells C_i, we are interested by polynomials $\mathcal{P}_i(\mathbf{x}, \bar{u})$ of degree k built on any cell i and of same mean on cell i as u :

$$\begin{aligned}&\mathcal{P}_i(\mathbf{x}, \bar{u}) = \sum_{m=0}^{k} \frac{1}{m!} \sum_{|\boldsymbol{\alpha}|=m} \binom{m}{\boldsymbol{\alpha}} (\mathbf{x} - \mathbf{c}_i)^{\boldsymbol{\alpha}} C^{\boldsymbol{\alpha}}, \quad \forall \mathbf{x} \in C_i, \\ &\int_{C_i} \mathcal{P}_i(\mathbf{x}, \bar{u}) dV = \bar{u}_i, \quad \mathit{and} \quad \int_{C_j} \mathcal{P}_i(\mathbf{x}, \bar{u}) dV = \bar{u}_j, \quad \forall\, j \in J(i)\end{aligned} \tag{5.1}$$

where $J(i)$ a set of cells close to cell C_i and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$ holds for the usual multi-index notation :

$$(\mathbf{x} - \mathbf{c}_i)^{\boldsymbol{\alpha}} = (x - c_i^x)^{\alpha_1} (y - c_i^y)^{\alpha_2} (z - c_i^z)^{\alpha_3}.$$

In practice, instead of (5.1), the least square version is considered :

Definition 2.1 *We call a least square reconstruction the solution of the minimisation problem :*

$$\begin{aligned}&\mathcal{P}_i^{LS}(\mathbf{x}, \bar{u}) = \sum_{m=0}^{k} \frac{1}{m!} \sum_{|\boldsymbol{\alpha}|=m} \binom{m}{\boldsymbol{\alpha}} (\mathbf{x} - \mathbf{c}_i)^{\boldsymbol{\alpha}} C^{\boldsymbol{\alpha}}, \quad \forall \mathbf{x} \in C_i, \\ &\int_{C_i} \mathcal{P}_i^{LS}(\mathbf{x}, \bar{u}) dV = \bar{u}_i, \quad \mathit{and} \\ &\mathbf{C} = Argmin \sum_{j \in J(i)} \left(\int_{C_j} \mathcal{P}_i^{LS}(\mathbf{x}, \bar{u}) dV - \bar{u}_j \right)^2\end{aligned} \tag{5.2}$$

where $\mathbf{C} = (C^{\boldsymbol{\alpha}}, |\boldsymbol{\alpha}| \leq m)$.

□

According to [43], for a sufficiently large neighborhood $J(i)$ of cells around i, we have :

$$\mathcal{P}_i^{LS}(\mathbf{x}, \bar{u}) = u(\mathbf{x}) + O(h^{k+1}).$$

when diameters of cells are less than h. In particular it implies that under the above condition P^{LS} is k-exact (mapping a kth-order polynomial in itself). In [4], the authors uses a result from [33] to give a more accurate estimate in the Sobolev space $W^{m,p}(\Omega)$ equipped with the following norm and semi-norm : $||.||_{m,p,\Omega}$:

$$||u||_{m,p,\Omega} = \left(\sum_{|\boldsymbol{\alpha}|=0}^{|\boldsymbol{\alpha}|=m} ||D^{\boldsymbol{\alpha}} u||_{p,\Omega}^p \right)^{\frac{1}{p}}, \quad |u|_{m,p,\Omega} = \left(\sum_{|\boldsymbol{\alpha}|=m} ||D^{\boldsymbol{\alpha}} u||_{p,\Omega}^p \right)^{\frac{1}{p}}.$$

It writes :

$$||u - R^k u||_{m,p,\Omega} \;\leq\; C|u|_{k+1,p,\Omega}\, \frac{h^{k+1}}{\rho^m} \tag{5.3}$$

for a certain constant C, and where ρ is related to the shape of cells and m is any integer such that $0 \leq m \leq k+1$. This shows that the reconstruction error is, when $k = 2$, effectively expressed in terms of the third derivative of u (fourth order when $k = 3$).

A last remark is that in the case of a least-square based reconstruction, if the number of cells of the support is exactly the number of unknown coefficients (and sufficiently well distributed in the space, e.g. not aligned), then the minimum of the least-square functional is zero which shows (for a smooth function) that the reconstruction is equal to the initial function in one point of each neighboring cell, in other words, the reconstruction is an interpolation, the error of which is given by the Taylor expansion. In the general case, we do not have a precise estimate and we choose to get inspired by the Taylor expansion and heuristically write our reconstruction error estimate as follows :

$$||u - R^k u(\mathbf{x})| \preceq \frac{1}{(k+1)!} \sup_{\delta\mathbf{x}} |D^{k+1}u(\delta\mathbf{x})^{k+1}| \,. \tag{5.4}$$

where $\preceq$ holds for an inequality which holds for mesh size sufficiently small and where the $\delta\mathbf{x}$ describes the local mesh sizes in all the space directions.

3 CENO for Euler and Navier-Stokes equations

3.1 Euler model

We write the unsteady Euler equations as follows in the computational domain $\Omega \subset \mathbb{R}^3$:

$$\Psi(W) = \frac{\partial W}{\partial t} + \nabla \cdot \mathcal{F}(W) = 0 \quad \text{in } \Omega, \tag{5.5}$$

where $W = {}^t(\rho, \rho u, \rho v, \rho w, \rho E)$ is the vector of conservative variables and $\mathcal{F}(W) = (\mathcal{F}_1(W)$,$\mathcal{F}_2(W)$,$\mathcal{F}_3(W))$ is the convective flux :

$$\mathcal{F}(W) = \left(\begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{pmatrix}, \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \end{pmatrix}, \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho E + p)w \end{pmatrix} \right) \tag{5.6}$$

so that the state equation becomes :

$$\frac{\partial W}{\partial t} + \frac{\partial \mathcal{F}_1(W)}{\partial x} + \frac{\partial \mathcal{F}_2(W)}{\partial y} + \frac{\partial \mathcal{F}_3(W)}{\partial z} = 0.$$

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

$$p = (\gamma - 1)\Big(\rho E - \frac{1}{2}\rho|\mathbf{u}|^2\Big), \tag{5.7}$$

where γ is constant. We also write (5.5) in variational form :

$$\begin{gathered} \text{Find } W \in \mathcal{V} \text{ such that } \ \forall\, \boldsymbol{\phi} \in \mathcal{V} : \\ \int_0^T \int_\Omega \Big(\boldsymbol{\phi}\frac{\partial W}{\partial t} + \boldsymbol{\phi}\nabla\cdot\mathcal{F}(u)\Big)\, d\Omega\, dt = \int_0^T \int_\Gamma \boldsymbol{\phi}\mathcal{F}_\Gamma(u)\, d\Gamma\, dt. \end{gathered} \tag{5.8}$$

Here $\mathcal{V} = [L^2\big(0,T;H^1(\Omega)\big) \cap H^1\big(0,T;L^2(\Omega)\big)]^5$ and $\boldsymbol{\phi} = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5)$ is a test function. As right-hand side we have an integral of the various boundary fluxes $\mathcal{F}_\Gamma$ for various boundary conditions, which we do not need to detail here. Defining :

$$B(W,\boldsymbol{\phi}) = \int_0^T \int_\Omega \Big(\boldsymbol{\phi}\frac{\partial W}{\partial t} + \boldsymbol{\phi}\nabla\cdot\mathcal{F}(W)\Big)\, d\Omega\, dt - \int_0^T \int_\Gamma \boldsymbol{\phi}\mathcal{F}_\Gamma(W)\, d\Gamma\, dt,$$

then a compact form of the variational formulation writes :

$$\text{Find } W \in \mathcal{V} \text{ such that } B(W,\boldsymbol{\phi}) \ = \ 0, \quad \ \forall\, \boldsymbol{\phi} \in \mathcal{V}. \tag{5.9}$$

3.2 k-exact finite volume

3.2.1 Conservative variable

We assume that the components of the solution W of (5.8) are polynomials of degree at most k. Let us assume that the computational domain Ω is partitioned into a set of i_{max} cells $C_i, i = 1, ..., i_{max}$. The semi-discrete in space finite-volume method is equivalent to considering as space-time test functions the product $\Phi_{i,t} = \chi_{C_i}\delta_t$ of the characteristic function χ_{C_i} of cell C_i by the Dirac function in time δ_t in such a way that $\int\int W\Phi_{i,t}\mathrm{d}x\mathrm{d}t = \int_{C_i} W(t)\mathrm{d}x$. After an integration by parts, the following holds for the exact solution for any cell C_i :

$$\int_{C_i} \frac{\partial W}{\partial t}\mathrm{d}\Omega + \int_{\partial C_i} \mathcal{F}(W)\cdot\mathbf{n}\mathrm{d}\sigma = 0 \ \ \forall\, t, \quad \forall\, i \tag{5.10}$$

where $\mathcal{F}$ is replaced by $\mathcal{F}_\Gamma$ on $\partial C_i \cap \Gamma$. The field W can be projected in a spatially-discrete field $\bar{W} = (W_1, .., W_{imax})$ defined by :

$$\bar{W}_i = \int_{C_i} W\mathrm{d}\Omega. \tag{5.11}$$

Let $\Big(\mathcal{P}_i\Big)_{i=1,...,imax}$ a set of polynomials defined on Ω.

Definition 3.1 *We say that the set of polynomial* $\Big(\mathcal{P}_i\Big)_{i=1,...,imax}$ *is a reconstruction of* $\bar{W}$ *if for any* $i = 1, ..., imax$*, we have*

$$\int_{C_i} \mathcal{P}_i d\Omega = \bar{W}_i. \tag{5.12}$$

A central property is the k*-exact reconstruction assumption* :

Definition 3.2 *A reconstruction*

$$\mathcal{P}_i^R : \bar{W} \mapsto \mathcal{P}_i^R(\bar{W})$$

at vertex i *is* k*-exact if for any polynomial* W *of degree* k*, for any* $\mathbf{x}$ *in* Ω*,*

$$\mathcal{P}_i^R(\bar{W})(\mathbf{x}) = W(\mathbf{x}). \tag{5.13}$$

☐

Lemma 3.1 *The least square reconstruction defined in (5.2) is a* k*-exact reconstruction if a sufficiently large set of neighboring cells is used.*

☐

Assuming that the whole set $\Big(\mathcal{P}_j(\bar{v})\Big)_{j=1,...,imax}$ is k-exact, we have $\mathcal{P}_j(\bar{W}) = W$ for any j. As a consequence we have :

$$\int_{C_i} \frac{\partial \mathcal{P}_j(\bar{W})}{\partial t} \mathrm{d}\Omega + \int_{\partial C_i} \mathcal{F}(\mathcal{P}_j(\bar{W})) \cdot \mathbf{n} \mathrm{d}\sigma = 0 \quad \forall\ t. \tag{5.14}$$

In order to build a spatially discrete unsteady problem with a spatially discrete unknown, we introduce a time-dependant vector $v_i(t)$ such that :

$$\bar{v}_i(0) = \int_{C_i} W(\mathbf{x}, 0) \mathrm{d}\Omega \ \ \forall\, i. \tag{5.15}$$

Vector $v = \Big(v_i(t)\Big)_{i=1,imax,t\in[0,T]}$ is the semi-discrete solution of the k-exact reconstruction finite volume :

Find v such that (5.15) holds together with :

$$\int_{C_i} \frac{\partial \mathcal{P}_i(\bar{v})}{\partial t} \mathrm{d}\Omega + \int_{\partial C_i} \mathcal{F}(\mathcal{P}_i(\bar{v})) \cdot \mathbf{n} \mathrm{d}\sigma = 0 \quad \forall\ t.$$

If we add the following *conservation assumption* concerning the reconstruction $\mathcal{P}$:

$$\int_{C_i} \mathcal{P}_i(\bar{v}) \mathrm{d}\Omega = meas(C_i)\bar{v}_i \ \ \forall\, i \tag{5.16}$$

We get a *finite-volume reconstruction formulation* :

$$\begin{aligned}&\text{Find } \bar{v}_i(t) \text{ such that (5.15),(5.16) holds together with :}\\ &\int_{C_i} \frac{\partial \bar{v}_i}{\partial t} \mathrm{d}\Omega + \int_{\partial C_i} \mathcal{F}(\mathcal{P}_i(\bar{v})) \cdot \mathbf{n} \mathrm{d}\sigma = 0 \quad \forall \ t, \forall \ i.\end{aligned} \tag{5.17}$$

Definition :*The discretization of a PDE is k-exact if for any exact polynomial solution v of the continuous PDE having a degree less or equal to k, its vector of cell-means $\bar{v}$ is solution of the discrete PDE.*

Proposition :*The finite volume spatial discretization (5.17) is a k-exact approximation of (5.9).*

In practice, we shall relax the above exactness of integrations as follows :

Proposition :*We keep k-exactness if we use in (5.17) numerical integrations on C_i and ∂C_i which are exact for polynomials W of degree at most k.*

Remark :*In practice we say that we have weak k-exactness if we use numerical integrations on C_i and ∂C_i which are exact for fluxes $\mathcal{F}(W)$ of degree at most k.*

3.3 Navier-Stokes model

The compressible Navier-Stokes equations for mass, momentum and energy conservation read :

$$\left\{\begin{array}{rcl} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &=& 0\,, \\ \dfrac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p &=& \nabla \cdot \mathcal{T}\,, \\ \dfrac{\partial (\rho E)}{\partial t} + \nabla \cdot ((\rho E + p)\mathbf{u}) &=& \nabla \cdot (\mathcal{T} \cdot \mathbf{u}) + \nabla \cdot (\lambda \nabla T)\,, \end{array}\right. \tag{5.18}$$

where ρ denotes the density (kg/m^3), $\mathbf{u}$ the velocity (m/s), E the total energy per mass ($m^2.s^{-2}$), p the pressure (N/m^2), T the temperature (K), μ the laminar dynamic viscosity ($kg/(m.s)$) and λ the laminar conductivity. $\mathcal{T}$ is the laminar stress tensor :

$$\mathcal{T} = \mu \left[(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) - \frac{2}{3} \nabla .\, \mathbf{u}\, I \right] \,,$$

where (in 3D) $\mathbf{u} = (u, v, w)$ and

$$\nabla.\,\mathbf{u}\,I = \begin{pmatrix} u_x + v_y + w_z & 0 & 0 \\ 0 & u_x + v_y + w_z & 0 \\ 0 & 0 & u_x + v_y + w_z \end{pmatrix},$$

where $u_x = \frac{\partial u}{\partial x}$, $u_y = \frac{\partial u}{\partial y}$, $u_z = \frac{\partial u}{\partial z}$ (idem for v and w). Finally the equation of state (5.7) and the following relation :

$$T = (E - \frac{1}{2}\rho\mathbf{u}^2)/(\rho C_v)$$

(where C_v is the heat capacity at constant volume) permits to close the system.

The RHS of (9.17) is written in the following compact way :

$$\mathcal{F}^{\mathrm{V}}(W) = [0, \sigma, -(-\lambda\nabla T - \mathbf{u}\cdot\sigma]^T \;\; ; \;\; \sigma = \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T) - \frac{2}{3}\mu\nabla\cdot\mathbf{u}I. \qquad (5.19)$$

The finite-volume reconstruction scheme analog to (5.17) writes :

Find $\bar{v}_i(t)$ *such that (5.15),(5.16) holds together with :*

$$\int_{C_i} \frac{\partial \bar{v}_i}{\partial t} \mathrm{d}\Omega + \int_{\partial C_i} \left[\mathcal{F}(\mathcal{P}_i(\bar{v})) - \mathcal{F}^{\mathrm{V}}(\mathcal{P}_i(\bar{v}))\right] \cdot \mathbf{n}\mathrm{d}\sigma = 0 \quad \forall\ t, \forall\ i. \qquad (5.20)$$

3.4 Vertex-CENO implementation

We choose a reconstruction-based finite-volume method, getting inspired by the unlimited version of the reconstruction technique of Barth [17] and of the CENO methods developed by Groth and co-workers, [65, 64, 55, 67, 66, 32], Olliver-Gooch and co-workers [97, 96, 101, 105, 14, 119, 135, 137, 136], and also [107] and [118]. Concerning the location of the nodes with respect to the mesh elements, we prefer to minimize the number of unknowns with respect to a given mesh and therefore we keep the vertex-centered location already successfully used for second-order anisotropic (Hessian-based or Goal-oriented) mesh adaptation [10, 23, 84]. For a more detailed description of the CENO approach presented here for scalar advection see [56]. The main features are :
(a) vertex centered,
(b) dual median cells around the vertex,
(c) a single mean square quadratic reconstruction for each dual cell
(d) Roe approximate Riemann solver for stabilization,
(e) explicit multi-stage time-stepping or implicit backward-Differencing Formula.

3.4.1 Finite Volume definition

The computational domain is divided in tetrahedra and in a dual tesselation in cells C_i around vertices i. Each cell C_i is limited by :
- possibly a portion of $\partial\Omega$, boundary of the computational domain,
- a set of m_{ij} triangular internal facets $(f_{ij}^m, m = 1, m_{ij})$ separating C_i and C_j, for any neighboring vertex j [2].

The vertices of any internal *facet* f_{ij}^m of $\partial C_i \cap \partial C_j \neq \varnothing$ has the following vertices (Figure 5.1) :
- the middle I_{ij} of edge ij,
- the centroid g_{ijk} of a tetrahedron face ijk for some vertex k, neighbor of i and j,
- the centroid G_{ijkl} of tetrahedron $ijkl$, for some vertex l, neighbor of i,j, and k.

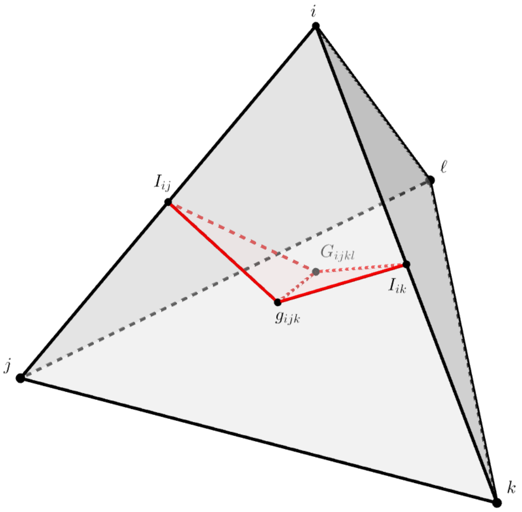


FIGURE 5.1 – Visualisation of the facet $I_{ij}g_{ijk}G_{ijkl}$ between $Cell_i$ and $Cell_j$.

We define the discrete space $\mathcal{V}_0$ of functions of $\mathcal{V}$ which are constant on any dual cell C_i.

3.4.2 Reconstruction

Let us define a *discrete reconstruction operator* R_k^0. The operator R_0^k reconstructs a function of $\mathcal{V}_0$ in each cell C_i under the form of a kth-order polynomial : $R_0^k u_0|_{C_i} = \mathcal{P}_i^k(\mathbf{x})$. Given the means $(\overline{u_{0,i}}, i = 1, ...)$ of u_0 on cells C_i, $\mathcal{P}_i^k(\mathbf{x})$ is defined by the $c_{i,\alpha}$, $|\alpha| \leq k$

---

[2] In practice internal facets can be gathered in groups of two coplanar facets

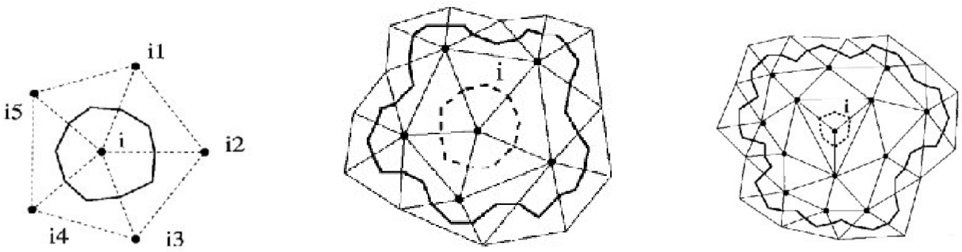


FIGURE 5.2 – Dual cell and two reconstruction molecules in 2D

such that :

$$\mathcal{P}_i^k(\mathbf{x}) = \overline{u_{0,i}} + \sum_{|\alpha|\leq k} c_{i,\alpha}[(\mathbf{x}-\mathbf{c}_i)^\alpha - \overline{(\mathbf{x}-\mathbf{c}_i)^\alpha}]$$

$$\overline{\mathcal{P}_{i,i}} = \overline{u_{0,i}} \quad ; \quad (c_{i,\alpha}, |\alpha|\leq k) \ = \ Arg\min \sum_{j\in N(i)} (\overline{\mathcal{P}_{i,j}} - \overline{u_{0,j}})^2$$

where $\overline{\mathcal{P}_{i,j}}$ stands for the mean of $\mathcal{P}_i^k(\mathbf{x})$ on cell j, and the set of neighboring cells is taken sufficiently large for an accurate quadratic reconstruction (see Fig. 5.2). In 3D this set involves generally 30 cells.

3.4.3 Euler fluxes

For the Euler model (5.9), the *semi-discretized CENO scheme* writes :

Find $u_0 \in \mathcal{C}^1([0,T];\mathcal{V}_0)$ such that $B_h(R_0^k u_0, v_0) \ = \ 0 \quad \forall\, v_0 \in \mathcal{C}^1([0,T];\mathcal{V}_0)$ with :

$$B_h(R_0^k u_0, v_0) \ = \ \int_{\Omega\times[0,T]} v_0 \frac{\partial R_0^k u_0}{\partial t}\, d\Omega \, + \\ \int_{\Omega\times[0,T]} v_0 \nabla_h \cdot \mathcal{F}(R_0^k u_0)\, d\Omega dt \ - \ \int_{\Gamma\times[0,T]} v_0 \mathcal{F}_\Gamma(R_3^0 u_0)\, d\Gamma\, dt \tag{5.21}$$

In this equation, the term $\nabla_h \cdot \mathcal{F}(R_3^0 u_0)$ needs to be defined, since $\mathcal{F}(R_0^k u_0)$ is discontinuous at cells interfaces. Taking v_0 as a characteristic function of cell C_i, we get a finite volume formulation :

$$\forall C_i, \ \int_{C_i} \frac{\partial R_0^k u_0}{\partial t}\, d\Omega \ + \ \int_{C_i} \nabla_h \cdot \mathcal{F}(R_0^k u_0)\, d\Omega \ - \ \int_{\partial C_i \cap \Gamma} \mathcal{F}_\Gamma(R_0^k u_0)\, d\Gamma \ = \ 0 \quad \forall t.$$

or (using $\overline{\mathcal{P}_{i,i}} = \overline{u_{0,i}}$) :

$$\forall C_i, \ \frac{\partial}{\partial t} \int_{C_i} u_0\, d\Omega \ + \ \int_{\partial C_i} \mathcal{F}(R_0^k u_0) \cdot \mathbf{n}\, d\Gamma - \int_{\partial C_i \cap \Gamma} \mathcal{F}_\Gamma(R_0^k u_0)\, d\Gamma \ = \ 0. \tag{5.22}$$

Neither the discrete divergence $\nabla_h \cdot$ nor the CENO interpolation are defined by the definition of the reconstruction. Indeed, the reconstruction performed in each cell produces a global field which is generally discontinuous at cell interfaces $\partial C_i \cap \partial C_j$, see Fig.5.4. In order to fix an integration value at the interface, we can consider an *arithmetic mean* of the fluxes values for the two reconstruction values :

$$\mathcal{F}(R_0^k u_0)^{mean}|_{\partial C_i \cap \partial C_j} \cdot \mathbf{n} = \frac{1}{2}\left(\mathcal{F}(R_0^k u_0)|_{\partial C_i} + \mathcal{F}(R_0^k u_0)|_{\partial C_j}\right) \cdot \mathbf{n} \tag{5.23}$$

where $(R_0^k u_0)|_{\partial C_i}$ holds for the value at cell boundary of the reconstructed $R_0^k u_0|_{C_i}$ on cell C_i. The above mean is computed on the four Gauss integration points $(\mathbf{g}_{ij}^{m,\alpha}, \alpha = 1, 4)$

Point location	Weight
Triangles	
$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$	$-\frac{9}{16}$
$(\frac{1}{5}, \frac{1}{5}, \frac{3}{5})$	$\frac{25}{48}$
$(\frac{1}{5}, \frac{3}{5}, \frac{1}{5})$	$\frac{25}{48}$
$(\frac{3}{5}, \frac{1}{5}, \frac{1}{5})$	$\frac{25}{48}$
Tetrahedra	
$(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$	$-\frac{4}{5}$
$(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{2})$	$\frac{9}{20}$
$(\frac{1}{6}, \frac{1}{6}, \frac{1}{2}, \frac{1}{6})$	$\frac{9}{20}$
$(\frac{1}{6}, \frac{1}{2}, \frac{1}{6}, \frac{1}{6})$	$\frac{9}{20}$
$(\frac{1}{2}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$	$\frac{9}{20}$

Table 5.1 – Quadrature for fourth-order CENO in triangles (for flux integration) and in tetrahedra.

of each facet $(f_{ij}^m), m = 1, ..., m_{ij}$ of $\partial C_i \cap \partial C_j$. The integration points are described in Table 1. Then an accurate definition of B_h is as follows :

$$\begin{aligned} B_h(R_0^k u_0, v_0) \ &= \ \int_0^T \Bigg\{ \sum_i \Bigg[\int_{C_i} v_0 \frac{\partial u_0}{\partial t} d\Omega + \int_0^T \int_{\partial\Gamma} v_0 \mathcal{F}_\Gamma(R_0^k u_0).d\Gamma dt \\ &+ \sum_{j\in\mathcal{N}(i)} \sum_{m=1,m_{ij}} \sum_{\alpha=0}^{\alpha=4} v_0(\mathbf{g}_{ij}^{m,\alpha}) \frac{\mathcal{F}(R_0^k u_0)|_{\partial C_i}(\mathbf{g}_{ij}^{m,\alpha}) + \mathcal{F}(R_0^k u_0)|_{\partial C_j}(\mathbf{g}_{ij}^{m,\alpha})}{8} d\sigma \Bigg] \Bigg\} dt. \end{aligned}$$

The discrete divergence involved in B is an extension of the continous one in the sense that, when applied to a continuous function $\mathcal{F}$, these operators provide the same image ($\nabla \cdot \mathcal{F} = \nabla_h \cdot \mathcal{F}$). In particular, the restriction of B_h to smooth function is B and for the continuous solution u we have :

$$B_h(u, v_0) = B(u, v_0) = 0 \quad \forall\, v_0 \in \mathcal{C}^1([0,T]; \mathcal{V}_0). \tag{5.24}$$

Remark 3.1 *In practice, the elementary interfaces in 3D are quadrilaterals, each quadrilateral grouping two coplanar facet $f_{ij}^{m_1}$, $f_{ij}^{m_2}$. Quadrilaterals can be integrated with four integration points.*

□

3.4.4 Viscous fluxes

The CENO viscous fluxes can be integrated as usually in finite-volume integrated at cell-interfaces with an *arithmetic mean* of the viscous fluxes values for the two cell-reconstruction values :

$$\mathcal{F}^{\mathbf{V}}(R_0^k u_0)^{mean}|_{\partial C_i \cap \partial C_j} \cdot \mathbf{n} = \frac{1}{2}\left(\mathcal{F}^{\mathbf{V}}(R_0^k u_0)|_{\partial C_i} + \mathcal{F}^{\mathbf{V}}(R_0^k u_0)|_{\partial C_j}\right) \cdot \mathbf{n} \tag{5.25}$$

where $(R_0^k u_0)|_{\partial C_i}$ holds for the value at cell boundary of the reconstructed $R_0^k u_0|_{C_i}$ on cell C_i. The same Gauss integration is then applied :

$$B_h^{\mathbf{V}}(R_0^k u_0, v_0) \;=\; \sum_{j \in \mathcal{N}(i)} \sum_{m=1,m_{ij}} \sum_{\alpha=0}^{\alpha=4} v_0(\mathbf{g}_{ij}^{m,\alpha}) \frac{\mathcal{F}^{\mathbf{V}}(R_0^k u_0)|_{\partial C_i}(\mathbf{g}_{ij}^{m,\alpha}) + \mathcal{F}^{\mathbf{V}}(R_0^k u_0)|_{\partial C_j}(\mathbf{g}_{ij}^{m,\alpha})}{8} d\sigma \Big] \Big\} dt. \tag{5.26}$$

However, in this study, we focus on LES and hybrid LES/RANS calculations for which viscous terms are very small. The main contribution of turbulence closure is the SGS term. Then in this first study, the FEM approximation existing in NiceFlow has been kept without change.

3.4.5 Time advancing

It remains to define a time discretization for (5.22). We can apply the standard explicit Runge-Kutta (RK4) time advancing. With the above central-diffferenced (5.23) spatial quadrature, this formulation produces a central-differenced numerical approximation which is fourth-order accurate in space and time for linear models. However in a strongly nonlinear setting, or on irregular unstructured meshes it cannot be used, due to a lack of stability. We use instead a Strong Stability Preserving scheme [122] of order three or four, which enjoy third and fourth order accuracy for nonlinear equations. With these explicit schemes, several high-order fluxes are evaluated at every time step, which results into a quite CPU consuming method. For large physical time computations like LES computations, a much more efficient option is to apply a Backward-Differencing time-advancing, BDF2, with the spatially first-order preconditioned unsteady Defect Correction method of [89] resulting in second order in time.

4 Lower-cost upwind CENO3

A direct implementation of an upwind CENO3 for compressible flows will produce a quite CPU consuming numerical software. One way to analyse the reasons of this high computational cost is to compare with a standard second-order MUSCL implementation. Then two sources of extra algorithmic complexity can be identified :
- for each finite-volume cell, a costly polynomial reconstruction is needed,
- for each flux between two neighboring cells, many Riemann solvers are computed for satisfying the accurate flux integration.
This section propose two approaches for reducing the two sources of complexity.

4.1 Lower-cost multiple-cell polynomial reconstruction

For each cell we need a reconstruction, but we can use one reconstruction for several cells by using the following proposition :

Proposition : *We keep k-exactness if the constrained optimum is replaced by the following two-step reconstruction :*

$$\begin{aligned} \text{Step 1 : } (c_{1,i,\alpha})_\alpha \;&=\; Arg\min \textstyle\sum_{j\in N(i)\cup\{i\}} (\overline{\mathcal{P}_{1,i,j}} - \overline{U_{1,j}})^2 \\ \text{Step 2 : } \mathcal{P}_{1,i} &= \mathcal{P}_{1,i} - \overline{\mathcal{P}_{1,i,i}} + \overline{U_{1,i}}. \end{aligned} \tag{5.27}$$

Proof : Indeed, for a degree k polynomial U_1, the first step will exactly find this polynomial, then $-\overline{\mathcal{P}_{1,i,i}} + \overline{U_{1,i}} = 0$ and the second step will not change it.□

The two-step reconstruction allows two different cells to have the same reconstruction molecule (choosing a unique reference point c_i), therefore solving Step 1 is done once for both, and solely Step 2 is done for each cell.

4.2 Unlimited lower cost upwind CENO3

In this section we concentrate on advective effects. Scheme (5.22) is usually combined with an approximate Riemann solver instead of the formulation (5.23) proposed in the previous section.

$$\Phi_{ij}^{CENO}(R_3^0 u_0, v_0) \;=\; \int_{\partial C_i \cap \partial C_j} \Phi^{Riemann}\Big(R_3^0 u_0|_{\partial C_i}, R_3^0 u_0|_{\partial C_j}, \mathbf{n}_{ij}\Big) d\sigma \tag{5.28}$$

This latter option produces the usual upwind-CENO scheme. The modification which we propose now is motivated by the seek for a less computer intensive scheme. It also permits a better control of the numerical viscosity. The Riemann solver is split as follows :

$$\begin{aligned}\Phi^{Riemann} &= \Phi^{central} + \Phi^{viscosity} \quad \text{with :}\\ \Phi^{central} &= \frac{1}{2}\Big(\mathbf{F}(R_3^0 u_0|_{\partial C_i})\cdot \mathbf{n}_{ij} + \mathbf{F}(R_3^0 u_0|_{\partial C_j})\cdot \mathbf{n}_{ij}\Big)\\ \Phi^{viscosity} &= \Phi^{Riemann} - \Phi^{central}.\end{aligned} \tag{5.29}$$

We keep the consistent central-differenced fluxes of scheme (5.22)-(5.23), which we integrate with the costly integration points of Section 3.4.3, giving the fluxes :

$$\begin{aligned}&\Phi_{ij}^{central}(R_3^0 u_0, v_0) =\\ &\sum_{f_{ij}^k \subset \partial C_i \cap \partial C_j} meas(f_{ij}^k) \sum_{\alpha=0}^{\alpha=4} \frac{\mathbf{F}\Big(R_3^0 u_0|_{\partial C_i}(\mathbf{g}_\alpha)\Big)\cdot \mathbf{n}_{ij} + \mathbf{F}\Big(R_3^0 u_0|_{\partial C_j}(\mathbf{g}_\alpha)\Big)\cdot \mathbf{n}_{ij}}{8}\end{aligned} \tag{5.30}$$

where f_{ij}^k is one of the four triangular facets the union of which is $\partial C_i \cap \partial C_j$ and $\mathbf{g}_\alpha^{ij}$ the integration points on f_{ij}^k. Since the resulting scheme is not stable, we need to add some numerical viscosity. This is done when adding the remaining part of the Riemann flux :

$$\Phi_{ij}^{viscosity}(R_3^0 u_0, v_0) = \int_{\partial C_i \cap \partial C_j} \Phi^{viscosity}\Big(R_3^0 u_0|_{\partial C_i}, R_3^0 u_0|_{\partial C_j}, \mathbf{n}_{ij}\Big) d\sigma \tag{5.31}$$

but, since this term does not contribute to the consistent approximation, it is possible, without degrading the scheme accuracy to evaluate the flux using a single integration point. A natural choice is the mid-edge I_{ij} of edge ij (Fig.5.4) :

$$\begin{aligned}&\Phi_{ij}^{low-cost-visc}(R_3^0 u_0, v_0) =\\ &\gamma_{low-cost-visc} meas(\partial C_i \cap \partial C_j)\Phi^{viscosity}\Big(R_3^0 u_0|_{\partial C_i}(I_{ij}), R_3^0 u_0|_{\partial C_j}(I_{ij}), \mathbf{n}_{ij}\Big).\end{aligned}$$

The coefficient $\gamma_{viscosity}$ is generally taken to unity, which give the usual CENO viscosity, or can be taken lower if low numerical viscosity is seeked, for example in combination with LES turbulence models.

Lemma 4.1 *The following flux is as accurate as CENO3 and is less expansive :*

$$\Phi_{ij}^{low-cost}(R_3^0 u_0, v_0) = \Phi_{ij}^{central}(R_3^0 u_0, v_0) + \Phi_{ij}^{low-cost-visc}(R_3^0 u_0, v_0). \tag{5.32}$$

The following flux is much less expensive than CENO3 but second-order accurate :

$$\begin{aligned}\Phi_{ij}^{o2}(R_3^0 u_0, v_0) &= meas(\partial C_i \cap \partial C_j)\\ \Phi^{central}&\Big(R_3^0 u_0|_{\partial C_i}(I_{ij}), R_3^0 u_0|_{\partial C_j}(I_{ij}), \mathbf{n}_{ij}\Big) + \Phi_{ij}^{low-cost-visc}(R_3^0 u_0, v_0).\end{aligned} \tag{5.33}$$

4.3 TVD-limited Lower cost upwind CENO3

As for the full-cost version, the lower cost upwind CENO3 is not a monotone approximation and can have stability problems near singularities or even extremas. We examine now how to make it more robuste, close to TVD.

It is first interesting to introduce the second-order version of CENO3. It consists in using a single integration point U_{ij} for integrating for vertex i (resp. U_{ji} for integrating for vertex j) between two neighboring vertices. In general the intermediate point of integration is the middle $I_{ij} = (i + j)/2$ of edge ij.

$$a_i \frac{dU_i}{dt} + \sum_{j \in V(i)} \Phi(U_{ij}, U_{ji}, \nu_{ij}) \ = \ 0 \tag{5.34}$$

where U_{ij} is the CENO3 interpolate using polynomial reconstruction in cell i, taken at location I_{ij}. Second-order CENO3 is an upwind scheme which is :

- much less computer expansive than the higher-order versions,
- can be made non-oscillatory by adding a TVD limiter.

In order to stabilize the second-order CENO3, we propose introducing the following TVD limiter :

$$a_i \frac{dU_i}{dt} + \sum_{j \in V(i)} \Phi(U_i + \frac{1}{2} L_{ij}(U), U_j - \frac{1}{2} L_{ji}(U), \nu_{ij}) \ = \ 0. \tag{5.35}$$

In order to define $L_{ij}(U)$ and $L_{ji}(U)$, we use the element T_{ij} *upstream* to vertex i with respect to edge ij if for any sufficiently small real number η the vector $-\eta\vec{ij}$ is inside element T_{ij}. Symmetrically, element T_{ji} is *downstream* to vertex i with respect to edge ij if for any small enough real number η the vector -$\eta\vec{ji}$ is inside element T_{ji}. Let us introduce the following notations :

$$\Delta^- U_{ij} \ = \ \nabla U\,|_{T_{ij}} . \vec{ij} \quad , \quad \Delta^0 U_{ij} = U_j - U_i \quad \text{a}nd \quad \Delta^- U_{ji} \ = \ \nabla U\,|_{T_{ji}} \cdot \vec{ij} \, ,$$

where the gradients are those of the P1 (continuous and linear) interpolation of U. We define :

$$L_{ij}(U) = L_{SB}(\Delta^- U_{ij} \, , \, \Delta^0 U_{ij} \, , \, \Delta^{HO} U_{ij}) \tag{5.36}$$

$$L_{ji}(U) = L_{SB}(\Delta^- U_{ji} \, , \, \Delta^0 U_{ij} \, , \, \Delta^{HO} U_{ji}) \tag{5.37}$$

where $\Delta^{HO} U_{ji}$ is a third way of evaluating the variation of U which we can introduce for increasing the accuracy of the resulting scheme. In the present work, $\Delta^{HO} U_{ji}$ is defineds in order to get either the second-order scheme or the CENO3 scheme, then recovering locally the full fourth-order accuracy for Euler model when no limiter applies. The limiter superbee L_{SB} proposed by Roe writes :

$$\begin{cases} L_{SB}(u,v,w) \ = \ 0 \qquad \text{if} \ \ uv \le 0 \\ \\ \qquad\qquad\quad\ = \ Sign(u) \ \min(\, 2\,|u| \, , \, 2\,|v| \, , \, |w|) \quad \text{otherwise.} \end{cases} \tag{5.38}$$

Under the above assumptions, the LED property can be established and in particular, the positivity of density for Euler discretizations holds :

Lemma 4.2 *The resulting TVD second-order CENO3 is second-order almost everywhere and LED.*

□

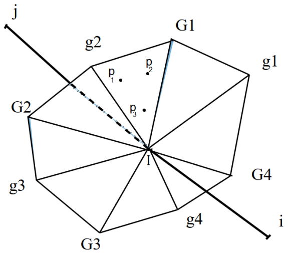


FIGURE 5.3 – Numerical quadrature for flux integration

5 A first numerical experiment

In order to illustrate the increment of accuracy brought by CENO3 with respect to the MUSCL-FEM approximation, we choose a test case close to the Norma benchmark test case, the turbulent flow around a cylinder. The incompressible inviscid flow around a cylinder presents an analytic output, the pressure coefficient at boundary. We choose a typical mesh used for the turbulent flow at Reynolds number 3900. It involves 1.8 M vertices and is refined near the cylinder. Views of sides of this 3D mesh are presented in Figure 5.4. In order to be close to the incompressible case, our computations are done with the compressible case by choosing the Mach number at farfield equal to 0.1 or 0.3. An extra difficulty is related with the Riemann solver which is used, some of them being not able to compute accurately very low Mach flows, e.g. the usual Roe approximate Riemann solver, some others being designed for it (Roe+LPMC approximate Riemann solver).

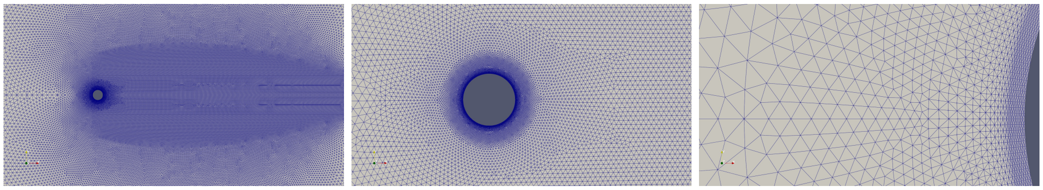


FIGURE 5.4 – Euler 3D flow past a cylinder : mesh

Figure 5.5 compares the computed surface pressures using the Roe-V6 and Roe-CENO3 schemes with the exact incompressible inviscid solution from potential theory given by :

$$C_p = 1 - 4\, sin^2(\theta)$$

The top figure clearly shows that the Roe-Ceno4 scheme is superior to the Roe-V6 scheme ; both of which do not use a low Mach number preconditionner (LMPC). However, when the LMPC is used with the Roe-V6, the differences between the surface pressures using the Roe-CENO3 and the Roe-V6+LMPC are not apparent to the naked eye in the top Figure. The differences will become clear in the middle and bottom figures which are zoom shots of the upstream and downstream regions.

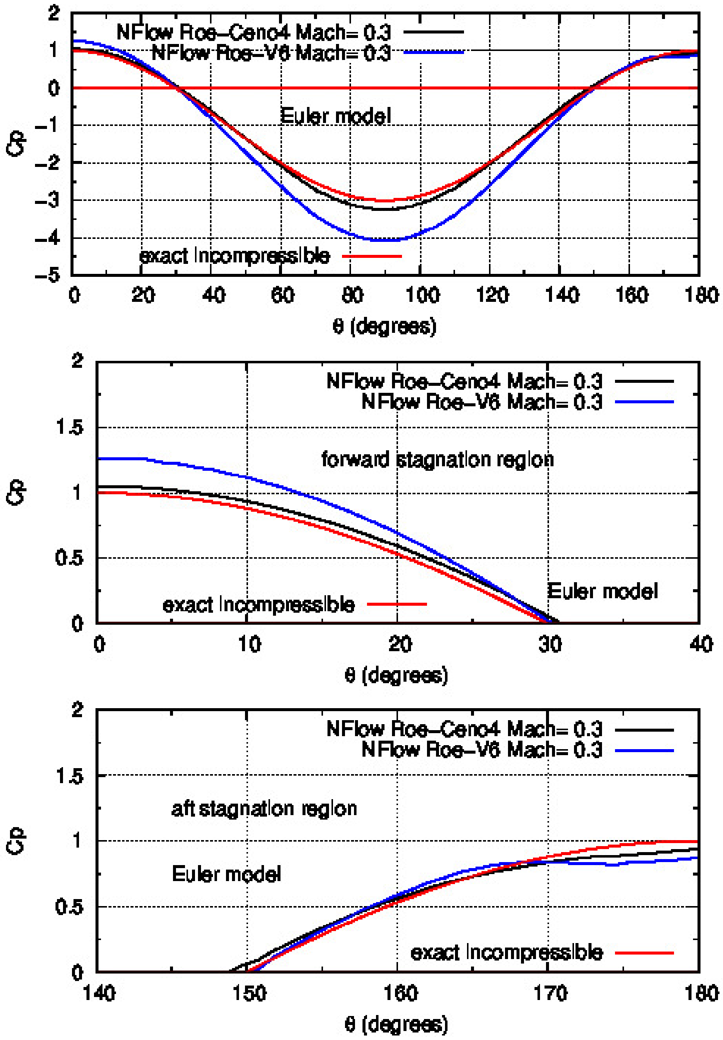


Figure 5.5 – Computed Cp mean pressure coefficients on the cylinder.

Note that the numerical Cp curves should start at a value of 1 for incompressible flow. This is not the case because our computations are at a Mach number of 0.3. To be able to do these computations at a smaller Mach number of say 0.1 we would need a low Mach preconditioning scheme, not yet available with CENO3.

The middle and bottom Figures 5.5 show the computed surface pressures in the upstream and downstream stagnation regions. What is surprising is that the Roe-CENO3 results agree well with the exact solution even though a low Mach preconditioner is not used. This has been remarked by many teams introducing higher order in their low Mach computations.

The middle Figure 5.5 shows a zoom of the upstream stagnation region in greater detail and the best agreement with the exact solution is obtained with the Roe-V6+LMPC scheme followed by the Roe-CENO3 scheme.

The bottom Figure 5.5 shows a zoom of the downstream stagnation region in greater detail. The Roe-CENO3 scheme gives the best agreement with the exact incompressible solution.

Table 5.2 gives the numerical mean CDrag coefficients, Cl', and the upstream/downstream stagnation pressure coefficients using the Euler model compared to the exact values.

Code	model	CDrag	Cl'	Cp-stag	Mach
Niceflow	Roe-V4	0.144	0.163	1.029/0.77	0.30
Niceflow	Roe-V4+LMPC	0.017	0.001	1.028/0.82	0.30
Niceflow	Roe-V6	0.032	0.019	1.264/0.89	0.30
Niceflow	Roe-V6+LMPC	0.027	0.014	1.032/0.70	0.30
Niceflow	Roe-Ceno4	0.011	0.008	1.051/0.94	0.3
exact	inviscid, inc	0	0	1/1	0

Table 5.2 – Numerical CDrag using the Euler model

This table shows that the NiceFlow Roe-CENO3 flux scheme is the most accurate.

Shown in Figure 5.6 are the Mach number contours using the Roe-CENO3 model, contours which present a rather good symmetry between right and left parts as expected for a potential flow.

6 Conclusion

The purpose of this chapter is to propose a higher-order approximation which is as compatible as possible with the existing finite volume schemes based on vertices of tetrahedrization, namely, (1) the MUSCL-FEM scheme with butterfly flux molecule (involving upwind and downwind tetrahedra) which is second-order accurate, and (2) the MUSCL-V6, also second order accurate, much less dissipative and enjoying supercon-

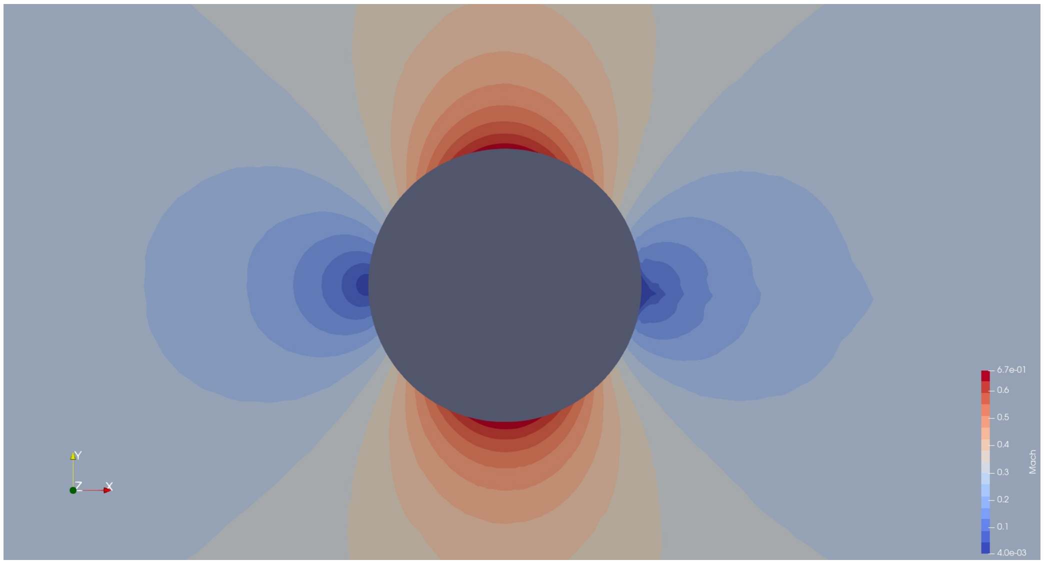


FIGURE 5.6 – Mach contours using Roe-CENO3 model.

vergence on Cartesian meshes.

A fourth-order accurate vertex centered CENO3-upwind has been defined through polynomial reconstruction on dual cells and high-order integration of Rieman-solver based fluxes with Gauss integration on cells interfaces. This approximation is upwind but not LED.
A less expansive but robust and fourth order accurate CENO3-MUSCL has been defined by combining :
- the CENO fluxes computed on the Gauss integration points with central differencing,
- the stabilization of the MUSCL or MUSCL-V6 built on mid-edges. It is not strictly LED, but still rather robust.

It should be noted that dissipation can be even further decreased by adding smoothness sensors dedicated to the inhibition of limiters in regions where the flow is regular.
The effect of approximating Euler fluxes with CENO3 is demonstrated with a first academic test case. This makes CENO3 an ingredient for improving our ability to compute the LES flows addressed in Norma and in this thesis. In Chapter III, we use CENO3 for the simulation of LES flows.

CHAPITRE 6

Computing the flow past a cylinder : influence of models and numerics

1 Introduction

During the work of the Norma consortium, some effort was devoted to the improvement of LES and hybrid simulations, and these improvements were evaluated with several workshop test cases, the more classical test case being the flow around a cylinder. In this chapter, we present computations predicting the flow around a circular cylinder, from sub-critical to super-critical Reynolds numbers. We discuss and compare several models and numerical approaches, with some emphasis on new mesh adaptation approach, and new high-order approximation as described in the previous chapters.

- Physics of the flow

The flow around a circular cylinder is an important prototype of flows that occurs in many engineering applications and in the environment. It is an interesting and important benchmark for CFD computations, because, although the geometry is simple, the physics of this flow is complex. It strongly varies with the Reynolds number and involves a rich variety of physical phenomena. These include laminar and turbulent boundary layers, flow separations due to adverse pressure gradients (not due to geometric singularities, which increases the difficulty of prediction), shear layers, laminar to turbulent transition, and shedding of vortex structures that are convected downstream and may be eventually broken up and diffused by turbulent motion. Different flow regimes can be distinguished depending on the range of Reynolds number (Re) considered : sub-critical, critical, super-critical and trans-critical (see, for example, [111, 140, 6]).

In the sub-critical regime ($10^3 < \mathrm{Re} < 2 \times 10^5$), the boundary layers separate in the laminar regime and the transition to turbulence occurs in the separated shear-layers. The separation occurs early on the wall, leading to a large wake and a high value of the drag coefficient of the order of 1.2, the Strouhal number being approximately 0.2.

For $2 \times 10^5 < \mathrm{Re} < 5 \times 10^5$, a critical regime is reached. The boundary layer remains laminar on one side of the cylinder. On the other side, the boundary layer is partly turbulent (transition to turbulence occured) and detaches further downstream from the obstacle, which results in a sudden decrease in the drag coefficient down to a minimum value of around 0.2. This phenomenon is known as drag crisis. Asymmetric forces acting on the cylinder surface are then observed with a non-zero mean lift coefficient. It is also seen that the Strouhal number increases sharply and that the separation angle increases up to 130-140 degrees.

In the super-critical regime ($5 \times 10^5 < \text{Re} < 2 \times 10^6$), the laminar to turbulent transition occurs in the boundary layer on each side of the cylinder, leading to a late separation and a thinner wake compared to the sub-critical case. The separation angle reduces from 140 degrees to 120 degrees, resulting in an increase of the drag coefficient with the Reynolds number.

The Reynolds number range $2 \times 10^6 < \text{Re} < 4 \times 10^6$ corresponds to the trans-critical regime. The boundary layers separate in the turbulent regime with a laminar-turbulent transition in the front part of the cylinder. The drag coefficient continues to increase and reaches a plateau up to a value of 0.5-0.6. It is also observed that the Strouhal number decreases.

- Previous calculations

The flow past a circular cylinder therefore constitutes an interesting and challenging test case for evaluating the performance of a turbulence model, especially if a wide range of Reynolds numbers is considered. RANS models, which are widely used, generally struggle to provide accurate predictions for flows with massive separation, as for instance flows around bluff bodies. An alternative approach is large-eddy simulation (LES), which is more accurate for massively separated flows but much computationally expensive than RANS. Indeed, the LES grid must be fine enough to resolve a significant part of the turbulent scales, and this becomes particularly critical in the regions close to the wall. In LES, it is necessary to compute the dynamics along a quite large time interval before accurate statistics can be obtained. Lastely, the cost of LES increases with the Reynolds number, in particular due to the turbulence of the boundary layer. In this context, hybrid strategies have been proposed in the literature, which combine the RANS and LES approaches (see [114, 48, 31, 90] for a review).

Among the publications dealing with the simulation of circular cylinder flows by hybrid methods, one can mention the work of Travin et al. [2] in which the Detached Eddy Simulations (DES) past a circular cylinder were performed in sub-critical and super-critical regimes. In this study, the inflow eddy viscosity is set to zero in the laminar-separation cases and to a non zero value in the turbulent-separation cases in order to manage the transition. Another interesting work is that of El Akoury et al. [7] in which, among the different turbulence models used for the simulation of a circular cylinder flow at Reynolds number 140.000, a DES/OES (Organised Eddy Simulation) and a DES based on an algebraic Reynolds stress model were applied, and for which the simulation results were compared to time-resolved PIV, phase-averaged fields and time-averaged wall pressure results.

Many publications deal with the simulation of the flow around a circular cylinder, but there are only few works in the literature investigating the drag crisis phenomenon using three-dimensional computational methods (some contributions can be found in [76, 110, 133, 129, 102, 24, 80, 138]), with results not always satisfactory mainly due to insufficiently accurate turbulence models. Among these publications dealing with the drag crisis, one can mention the work of Lehmkuhl et al. [76, 110] whose numerical results obtained with LES on fine meshes can be considered as reference numerical data due to the detailed analysis performed of the simulated flows and the prediction accuracy achieved.

- Influence of model

The subcritical cylinder flows, due to their laminar boundary layer, should not be computed with the statistical models which assume that the boundary layers are fully turbulent. Instead, Large Eddy Simulation (LES) can be used. The seminal Smagorinsky model can be successful for the turbulent wake but is not able to represent the laminar boundary layer (nor the turbulent one). Its dynamic adaptation in [51, 78] controls the behavior of the subgrid scale (SGS) coefficient near the boundary and will perform accurately. Another standpoint for this issue is a different SGS term introduced in the WALE model [98] and enjoys also the adequate behavior near boundary. A third methodology, the variational multiscale (VMS) [61, 73] focusses on restricting the LES filter to an approximation subspace and at the same time also enjoys the adequate behavior near boundary. Because the modeling principle and properties of these models are somewhat complementary, they can be combined into VMS-WALE or Dynamic-VMS [94] with measurable benefits.

The LES approach for the turbulent boundary layers of supercritical cylinder flows is of prohibitive computational cost, while statistical treatment does a rather good job with a much smaller computing effort. Indeed since boundary layer are more or less stably turbulent, the RANS option is reasonable. But most RANS models excessively damp most wake vortices, which deteriorates the accuracy of the prediction, in particular out from the near wall region. Then DDES and more generally hybrid RANS/LES models (both introduced in Chapter 3) seem to be a better answer.

Regarding transcritical flows, the flow configuration where the boundary layer is laminar on one side of the cylinder and partly turbulent on the other side, leads to consider that transition modeling is mandatory. We describe in Section 2 the transition model based on an intermittency equation introduced in [93] during the work of consortium Norma.

- Influence of numerics :
– discussion on LES with second-order numerical truncation
While, by construction, the filter size of LES, Δ_f should be larger than mesh size, Δ_g, in order to approximate accurately the non-neglected scales, the research of the lowest computational cost motivates the practitioner to set

$$\Delta_f = \Delta_g,$$

with the consequence that the smallest unfiltered scales are the smallest scales computed on the grid and are then very poorly approximated, whatever the accuracy of the numerical scheme.

When using a second-order accurate approximation, an important disadvantage comes from the fact that many LES models of Smagorinsky type are similar to second-order accurate truncation terms, in such a way that approximation errors are of same order as the filter model. According to an analysis of Ghosal [52] and to the outputs of many numerical computations, see e.g. [74], using a second-order accurate approximation may result in errors larger than the effect of LES modelling. It is then important to examine the impact of using a hiher order approximation. It remains that second-order accurate approximations are much used and very useful for computing LES flows in engineering. A good practice for increasing the confidence in second-order accurate LES computations is to compare (a) the LES-based computations with (b) their no-model counterpart in order two, see e.g. [94].

– **Problematics in LES/adaptation combination**
In industrial applications of LES and hybrid RANS/LES (including DES) models, local mesh adequation is challenged by different flow behavior, from thin laminar boundary layers to turbulent wakes with different scales of vortex shedding. Then the interest for the application of automated mesh adaptation to these complex flows is strong. An important issue is the timestep length adaptation. Indeed explicit time advancing is generally not used. When using an explicit time-advancing, a compromise should be found between the spatial error and the temporal one. Several propositions can be found in the litterature, [36, 49, 70, 87, 88]. An approach for time and space adaptation which is compatible with the anisotrope continuous mesh approach is addressed in Chapter 8.

But it seems today that the most difficult issue remains in relation with the spatial error evaluation and control. RANS adaptive methods, in particular those using anisotropic metrics in the continuous mesh approach have shown a certain success, see [8, 39].

In contrast, for the mesh error related with LES modeling, many attempts/experiments have been published without a definitive success, see for example [19, 74, 59, 46, 95, 25], except, probably the rather sound analysis of Toosi and Larssen [130], an analysis which we examine in Chapter 8.

In this chapter we restrict our mesh adaptive study to a spatial adaptation with a mesh frozen during the time-interval and we will try to show that already some advantages can be obtained from this standpoint.

2 Models

The models used in this chapter involve three RANS models (Spalart-Allmaras, $k-R$, $k-\varepsilon$ of Goldberg), a RANS with intermittency, several LES (Wale, Smagorinsky, VMS, dynamic), DDES and hybrid. *This chapter is a version of a preprint which presents theses models and some numerics, which are essentially already presented in the first chapters 2,3,4* *of the present thesis. The reader is invited to pass to test cases computions, starting at Section 4.*

2.1 RANS modelling

First, we want to specify that RANS stands for unsteady RANS throughout the document. In this work, and as far as the closure of the RANS equations is concerned, three low Reynolds RANS models are used, namely the Spalart-Allmaras (SA) model [125], the $k-\varepsilon$ model proposed in Goldberg et al. [53] (KEG) and the $k-R$ model (KR) recently introduced in [141] by Zhang et al. These models were chosen, especially the last two mentioned, because of their abilities to correctly predict separated flows with adverse pressure gradients that interest us in this study. SA is the standard basis of DES and DDES models. KEG was designed to improve the predictions of the standard $k-\varepsilon$ one for adverse pressure gradient flows, including separated flows. For the sake of brevity, the equations of these two models are not recalled in this document. KR was designed with the aim of correctly capturing non-equilibrium flows with separation and adverse pressure gradient. This recent two-equations RANS model is defined by the following transport equations on k and $R = k^2/\varepsilon$:

$$\begin{aligned} &\frac{\partial \rho k}{\partial t} + \mathrm{div}(\rho \mathbf{u} k) = \mu_t \tilde{S}^2 + \mathrm{div}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right] - \rho \frac{k^2}{R}, \\ &\frac{\partial \rho R}{\partial t} + \mathrm{div}(\rho \mathbf{u} R) = c_1 T_t \mu_t \tilde{S}^2 - \min\left(\rho c_2 k, \mu_t \frac{v}{a_1}\right) + \mathrm{div}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\nabla R\right] \\ &+ \left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{4}{k}\nabla k \cdot \nabla R - \left(f_d\,\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{2R}{k^2}\|\nabla k\|_2^2 - \left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{2}{R}\|\nabla R\|_2^2. \end{aligned} \tag{6.1}$$

where $\tilde{S}$ denotes a modified mean strain-rate (see [141] for its expression) and μ_t is the turbulent viscosity defined by

$$\mu_t = \rho c_\mu f_\mu \Big[\underbrace{k T_t (1 - f_c)}_{\mu_t^{(1)}} + \underbrace{R f_c}_{\mu_t^{(2)}} \Big] \tag{6.2}$$

in which c_μ, f_μ and T_t are given in [141], and f_c is an hybridization function defined by $f_c = \tanh\left(\max\left(0, \frac{s}{v} - 1\right)\right)$ with $s = \sqrt{2\sigma : \sigma}$ and $v = \sqrt{2\Omega : \Omega}$, σ being the strain tensor and Ω the vorticity tensor. It is worth noting that f_c is a function which tends to zero in the rotational region and unity in the irrotational region, switching between $\mu_t^{(1)}$ and $\mu_t^{(2)}$ in expression (6.2), $\mu_t^{(2)}$ being more suitable in shear regions.

2.2 Transition k-eps

We shall refer in tables to $k-\varepsilon-T$ model : the transition extension of the $k-\varepsilon$ model of Goldberg has been introduced in [93]. We describe this new model.
Let $P_k=\tau:\nabla\mathbf{u}$ and $D_k=\rho\varepsilon$ be the turbulent kinetic energy production and destruction terms. Let us define the laminar and turbulent flow regions in the boundary layer :

$$\begin{aligned} L &= \{\mathbf{x}\in\Omega_f \mid Re_\theta(\mathbf{x}) < Re_{\theta,S}(\mathbf{x})\},\\ Tu &= complement(L), \end{aligned}$$

where Re_θ denotes the Reynolds number based on the boundary layer thickness which is defined by $Re_\theta = 0.664\sqrt{Re|x|}$, and :

$$Re_{\theta,S} = 163+\exp\left(F_\lambda - \frac{100F_\lambda}{6.91}\sqrt{\frac{2}{3}k}\right).$$

with

$$F_\lambda = \begin{matrix} 6.91+2.48\lambda-12.27\lambda^2, & \lambda>0,\\ 6.91+2.48\lambda+63.64\lambda^2, & \lambda<0. \end{matrix}$$

and

$$\lambda = 0.664^2\frac{1}{Re}\frac{\partial u_x}{\partial x}(x-x_0)$$

where x_0 denotes the abscissa of the front body point. Following the work of Menter et al. (2015) and the works of Akhter et al. (2007, 2009, 2015), the present transition model is defined as follows :

$$\frac{\partial\rho k}{\partial t}+\mathrm{div}(\rho\mathbf{u}k) = \widetilde{P_k}-\widetilde{D_k}+\mathrm{div}\left[(\mu+\mu_t\sigma_k)\nabla k\right]$$

$$\frac{\partial\rho\varepsilon}{\partial t}+\mathrm{div}(\rho\mathbf{u}\varepsilon) = (c_{\varepsilon_1}P_k-c_{\varepsilon_2}D_k+E)\,T_t^{-1}+\mathrm{div}\left[(\mu+\mu_t\sigma_\varepsilon)\nabla\varepsilon\right]$$

$$\frac{\partial\rho\gamma}{\partial t}+\mathrm{div}(\rho\mathbf{u}\gamma) = c_{g_1}\gamma(1-\gamma)\frac{P_k}{k}+\rho c_{g_2}\frac{k^2}{\varepsilon}\nabla\gamma\cdot\nabla\gamma+\mathrm{div}\left[\sigma_\gamma(\mu+\mu_t)\nabla\gamma\right].$$

The intermittency model interacts with the turbulence model by modifying the turbulent kinetic energy equation. The new production and destruction terms are defined by :

$$\widetilde{P_k} = \begin{cases} 0 & \text{if } \mathbf{x}\in L,\\ \max(\gamma,\gamma_1)\,P_k & \text{otherwise.} \end{cases}$$

$$\widetilde{D_k} = \begin{cases} 0 & \text{if } \mathbf{x}\in L,\\ \max(\gamma,\gamma_2)\,D_k & \text{otherwise.} \end{cases}$$

where the model constants are defined by $\gamma_1 = 0$ and $\gamma_2 = 0.1$. From the above equations, one can notice that the baseline RANS model is recovered for an intermittency value $\gamma = 1$ (fully turbulent mode). A zero normal flux is also imposed on γ at the wall.

2.3 LES-WALE and VMS/ VMS-WALE

We refer to Chapter 3 for a description of these LES models.

2.4 DVMS/Dynamic

The Variational Multiscale (VMS) model for the large eddy simulation of turbulent flows has been introduced in [61] in combination with spectral methods. In [73], an extension to unstructured finite volumes is defined. In the present work, this method is integrated in the hybridization strategy for the closure of the LES part. Let us explain this VMS approach in a *simplified context*. Suppose the mesh is made of two embedded meshes. On the fine mesh we have a P^1-continuous finite-element approximation space V_h with the usual basis functions φ_i vanishing on all vertices but vertex i. Let V_{2h} represents its embedded coarse subspace. Let V_h' be the complementary space : $V_h = V_{2h} \oplus V_h'$. The space of *small scales* V_h' is spanned by only the fine basis functions φ_i' related to vertices which are not vertices of V_{2h}. Let us denote the compressible Navier-Stokes equations by : $\dfrac{\partial W}{\partial t} + \mathrm{div} F(W) = 0$ where $W = (\rho, \rho\mathbf{u}, E)$ are the flow variables, ρ being the density, $\mathbf{u}$ the velocity vector and E the total energy per unit volume.
The VMS discretization writes for $W_h = \sum W_i \, \varphi_i$:

$$\left(\frac{\partial W_h}{\partial t}, \varphi_i\right) + (\mathrm{div} F(W_h), \varphi_i) = -\left(\tau^{LES}({W_h}'), {\varphi_i}'\right) \tag{6.3}$$

where F denotes the convective and diffusive fluxes, and ${W_h}'$ represents the small scale component of the resolved flow variables W_h. For a test function related to a vertex of V_{2h}, the RHS vanishes, which limits the action of the LES term to small scales. In practice, embedding two unstructured meshes V_h and V_{2h} is a constraint that we want to avoid. The coarse level is then built from the agglomeration of direct neighboring cells. It remains to define the modeling term $\tau^{LES}(W_h')$. This term represents the subgrid-scale (SGS) stress term, acting only on the small revolved scale component W_h', and computed from the small resolved scale component of the flow field by applying either a Smagorinsky [124] or a WALE [98] SGS model, the constants of these models being evaluated by the Germano-Lilly dynamic procedure [51, 78]. The resulting model, for which a detailed description can be found in [94, 73], is denoted DVMS in this paper. It has been checked that combining the VMS approach and the dynamic procedure effectively brings improved predictions [94].

A key property of the VMS formulation is that the modeling of the dissipative effects of the unresolved structures is only applied on the small resolved scales. This property is not satisfied by LES models which also damp the large resolved scales. Important consequences are that a VMS model introduces less dissipation than its LES counterpart (based on the same SGS model) and that the backscatter transfer of energy from smallest scales to large scales is not damped by the model. The VMS approach then generally allows better behavior near walls, in shear layers and in the presence of large coherent structures. Moreover, in this work, the dynamic procedure, which provides a tuning of the SGS dissipation in space and time, is combined with the VMS approach, which limits its effects to the smallest resolved scales, so that the resulting DVMS model ensures synergistic effects.

2.5 DDES

The DDES approach ([127]) allows to delay an early transition from RANS to LES in the boundary layer by modifying the length scale d of the DES model. For this purpose, an adimensionalized fonction f_{ddes}, called the delaying function, is introduced :

$$f_{\text{ddes}} = 1 - \tanh((8r_d)^3) \quad \text{with} \quad r_d = \frac{\nu_t + \nu}{\max\left(\sqrt{\nabla \mathbf{u} : \nabla \mathbf{u}}\,, 10^{-10}\right) \kappa^2 d_w^2},$$

where κ denotes the von Kàrmàn constant ($K = 0.41$), d_w is the normal distance to the wall ν_t and ν are the turbulent kinematic viscosity and the fluid kinematic viscosity, respectively. For the DDES/Spalart-Allmaras version (DDES-SA), the length scale is then redefined as follows :

$$\tilde{d} = d - f_{\text{ddes}}\ \max(0, d - C_{DES}\Delta) \quad ; \quad C_{DES} = 0.65$$

and introduced in the closure equation :

$$\frac{D\tilde{\nu}}{Dt} = C_{b1}\tilde{S}\tilde{\nu} - c_{w1} f_w (\frac{\tilde{\nu}}{\tilde{d}})^2 + \frac{1}{\sigma}\Big[\text{div}[(\nu + \tilde{\nu})\nabla\tilde{\nu}] + c_{b2}\nabla\tilde{\nu}\cdot\nabla\tilde{\nu}\Big].$$

The DDES/two-equation models are obtained by replacing, in the ρk transport equation, the dissipation term by a DDES dissipation introducing the characteristic length l_{ddes} :

$$\frac{\partial \rho k}{\partial t} + \frac{\partial(\rho \tilde{v}_j k)}{\partial x_j} = \frac{\partial\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]}{\partial x_j} + \tau_{ij}\frac{\partial v_i}{\partial x_j} - \rho\frac{k^{3/2}}{l_{\text{ddes}}}$$

where $l_{\text{ddes}} = \dfrac{k^{3/2}}{\varepsilon} - f_{\text{ddes}}\ \max(0, \dfrac{k^{3/2}}{\varepsilon} - C_{DES}\Delta)$
and for the $k - R$ model, $\varepsilon = k^2/R$.

2.6 Hybrid strategies

Our hybrid strategies are based on a blending of either the RANS or DDES model with the DVMS approach. After semi-discretization, the hybrid equations can be written as :

$$\left(\frac{\partial W_h}{\partial t}, \varphi_i\right) + (\mathrm{div} F(W_h), \varphi_i) = -\theta \left(\tau^{mod}(W_h), \varphi_i\right) - (1-\theta)\left(\tau^{DVMS}(W'_h), \varphi'_i\right)$$

where the stress tensor τ^{mod} is defined by first choosing one of the three above RANS model, then τ^{mod} holds for either the RANS stress tensor or the stress tensor for a DDES model built with the chosen RANS approach. τ^{DVMS} is the SGS term that applies in DVMS on the small resolved components W'_h of the hybrid variable W_h, φ_i denotes the basis and test functions, F denotes the convective and viscous fluxes, and θ is the blending function.

The blending function plays a key role in our hybrid models since, according on its value, the model behaves like a RANS (or DDES) model, like a DVMS model, or between these two modes. This function must be able to allow an automatic and progressive switch from RANS to DVMS where the grid resolution is fine enough to resolve a significant part of the local turbulence scales or fluctuations, i.e. computational regions suitable for DVMS computations. Additionally, this blending function should prevent the activation of the DVMS model in the boundary layer, which would otherwise require a too fine mesh in order to obtain a good prediction of this region of the flow. Typically, it is desirable to use the RANS (or DDES) model around the body and the DVMS model elsewhere (if the fineness of the mesh allows it). DVMS should be applied in the wake region in order to more accurately predict the propagation of the fluctuations of the flow like those of pressure in the case of an aeroacoustic calculation. Indeed, the low dissipation introduced by the DVMS approach reduces the damping of such turbulent structures.

In this work, two blending functions are applied which can write as $\theta = 1 - f_d(1 - \tanh(\xi^2))$ with $\xi = \frac{\Delta}{l_{RANS}}$ where l_{RANS} denotes the characteristic RANS scale ($l_{RANS} = \frac{k^{3/2}}{\varepsilon}$), Δ is the filter width (defined in this study as the cubic root of the grid element volume), and f_d is a shielding function characterizing each of the two blending functions :

- *Option 1* : $f_d = f_{\text{ddes}}$ the delaying function used in DDES

- *Option 2* : $f_d = f_{\text{geo}} = \exp\left(-\frac{1}{\epsilon}\min^2(d-\delta_0, 0)\right)$ with d the normal distance to the wall, $\epsilon > 0$ small enough and $\delta_0 > 0$ (of the order of the boundary layer thickness).

3 Numerics

3.1 MUSCL and molecule extensions

The numerical approach selected in this work is basically a second-order accurate vertex centered approximation on tetrahedrizations. Dual cells are limited by facets, which are triangles, each of them being formed by a mid-edge, a centroid of face, a centroid of an element. The assembly of fluxes is edge based. For each edge ij of the mesh, fluxes are computed through the union of facets being the common boundary between two dual cells around the vertices i and j. The flux between each couple (i,j) of cells is numerically integrated through a unique approximate Riemann solver ARS$(W_{ij}, W_{ji}, \nu_{ij})$, in practice the Roe scheme. The integration values W_{ij} W_{ji} are two special *upwind and downwind reconstructions* of the unknown field W located at the middle I_{ij} of the edge ij. For these reconstructions, we consider two methods :
- the usual MUSCL method resulting in second order accuracy,
- the V6 scheme which has been introduced in [38, 26].

The interpolation W_{ij} uses the value of the field and its gradient on vertex j and on the vertices of the upwind tetrahedron T_{ij}. In the AIRONUM code, the scheme is fifth-order accurate on certain Cartesian meshes, and is (solely) stabilized by a sixth-order dissipation the strength of which can be tuned by a parameter $\gamma \in]0, 1]$. See [38, 26] or the monograph [39] for details. A very efficient similar method can be found in [16].

3.2 CENO3

The high-order scheme which we develop and test is a k-exact finite-volume ($k = 3$) of the family initiated in [18] and further developed in [55, 101, 107, 118]. The finite-volume which we use is of dual type, built with median plans and vertex-centered, i.e. strictly identical to the MUSCL of previous section 3.1. Means on neighboring cells $C_i, C_{i_1}, C_{i_2}, ...$ of a cell C_i around vertex i are used for reconstructing as a third degree polynomial $\mathcal{P}_i$ the variation of the unknown inside cell C_i, including its boundary. Then inviscid and viscous fluxes between cells are computed at cells boundary, with geometric means between boundary sides for viscous terms, and approximate Riemann solvers for inviscid terms. The resulting accuracy for inviscid terms is fourth order.

3.3 Anisotropic mesh adaptation

The capture of high Reynolds number flows with a mesh adaptation algorithm demands an algorithm able to find the boundary layer when starting from a uniform flow. This is obtained by using (1) an adequate *adaptation criterion* and (2) a stable and convergent *fixed point* for the coupling of flow and mesh. Main principles for a successful adaptation can be found in [39]. The transient fixed point (TFP) was introduced and discussed in [10] and [13].

In the approach chosen here, we restrict to LES flows for which a global shedding period T_s can be estimated, and we use kT_s as the size of the adaptation time interval, with $k = 2$. The TFP adaptation algorithm is performed in the following way for each of our computations. Due to quasi-periodicity, we choose to remesh on each period $2T_s$ by making only one fixed-point iteration. This means, as shown in Figure 6.1, that each of our sub-intervals $[t_{i-1}, t_i]$ is worth $2T_s$ and that once a first computation has been performed on this sub-interval, we generate a new mesh adapted to our flow (with respect to the chosen sensor) and recalculate the solution of this sub-interval on our new adapted mesh. Saying that we are only doing one fixed-point iteration means that we are only doing this procedure once. Quasi-steady features like boundary layers are accurately followed by the mesh, while unsteady vortices travel in a rather uniformly refined region.

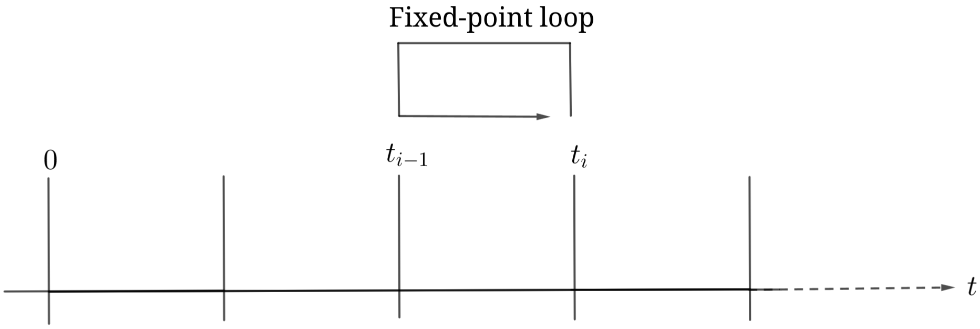


FIGURE 6.1 – Transient Fixed Point loop.

4 Subcritical 3900

When the Reynolds number is 3900, the boundary layer is laminar, but the separation shear layer is unstable and the wake is turbulent.

4.1 VMS-WALE, MUSCL, Adaptive

As mentioned previously, due to the quasi-periodicity of the flows, we chose to adapt our meshes over a time interval corresponding to a certain number of vortex shedding. Considering that we have an experimental Strouhal number of 0.2, we choose to remesh every $10\,s$, which corresponds to two periods, specifying that the flow velocity at infinity is $1\,m/s$. The anisotropic criterion is the Hessian of the Mach number. The final mesh after several fixed points , shown in Figure 6.2, has around $903\,K$ vertices.

We obtain a good prediction of the mean C_p distribution on cylinder wall, Figure 6.3. Two sets of figures, Figure 6.4 and Figure 6.5 show cuts of the instantaneous velocity fields and vorticity. The wakes are reasonably discretized. Lastly, we present 3D views of the Q-criterion, also showing that the main structures are captured at 5-10 diameters of the obstacle.

The overall bulk coefficients are relatively well predicted (Table 6.1). This illustrates the rather good performance of the mesh adaptive algorithm.

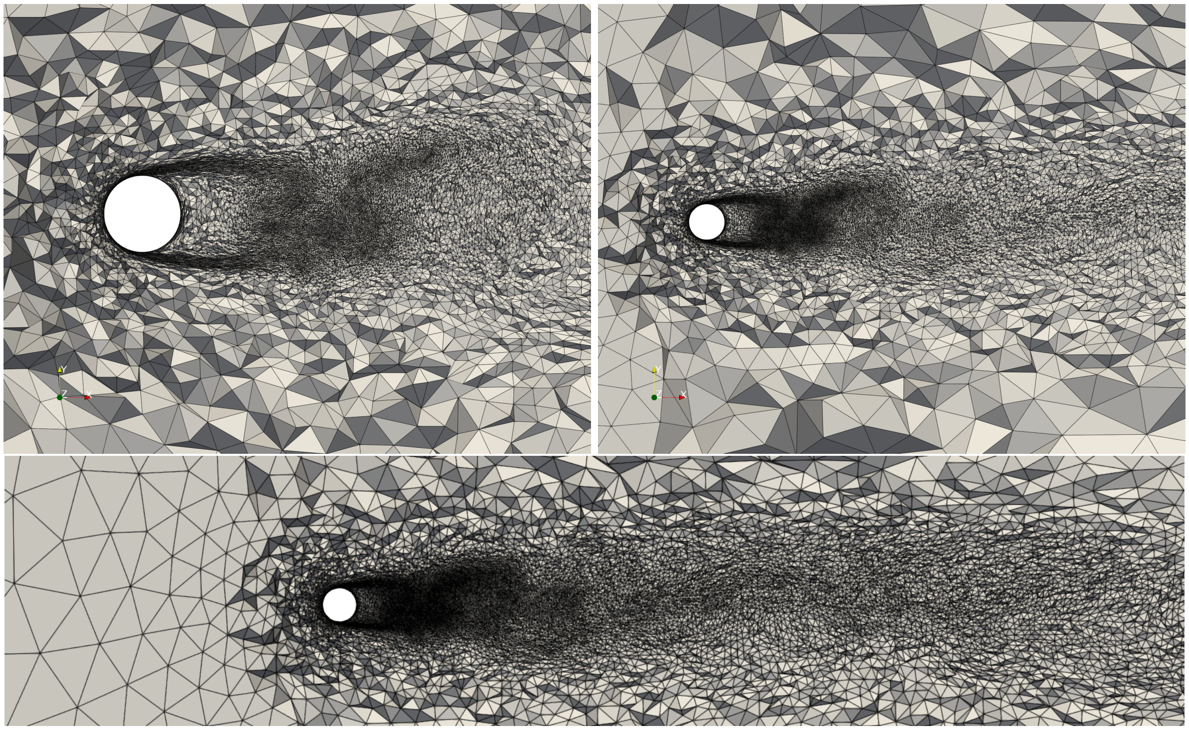


FIGURE 6.2 – Cylinder $Re = 3900$ VMS-WALE mesh adaptive simulation results : final mesh in cross-section.

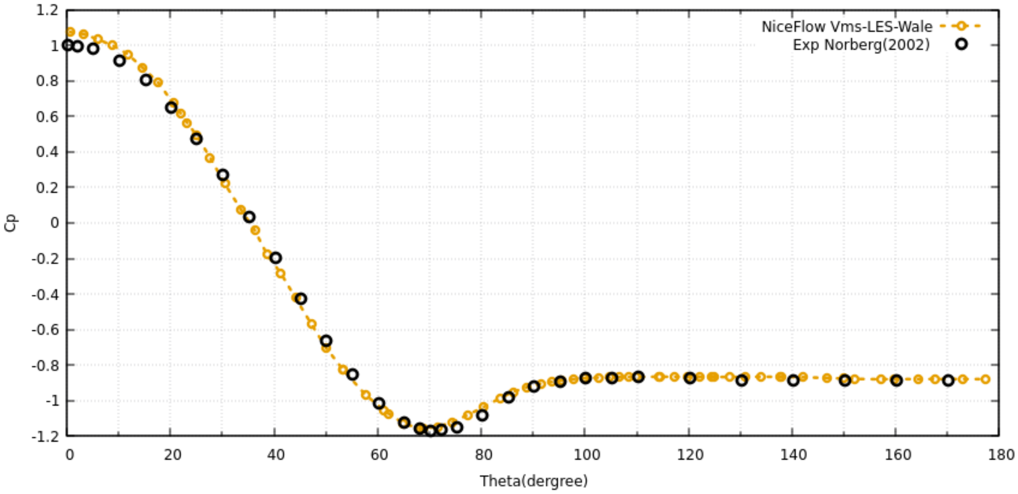


FIGURE 6.3 – Cylinder $Re = 3900$ VMS-WALE mesh adaptive simulation results : distribution over the cylinder surface of the mean pressure coeficient compared with the experimental measures of Norberg [100]. The C_p curves is slightly under $1.$ at stagnation point because this flow has been computed at farfield Mach number 0.3 (instead of being incompressible as the experimental flow).

Remark 4.1 *Note that the Cp curve should start at a value of* 1*. This is not the case because our computation is at a Mach number of* 0.3 *and not* 0.1 *. To be able to do this computation at a Mach number of* 0.1 *we would need a low Mach numerical scheme, which is not the case in NiceFlow®.*

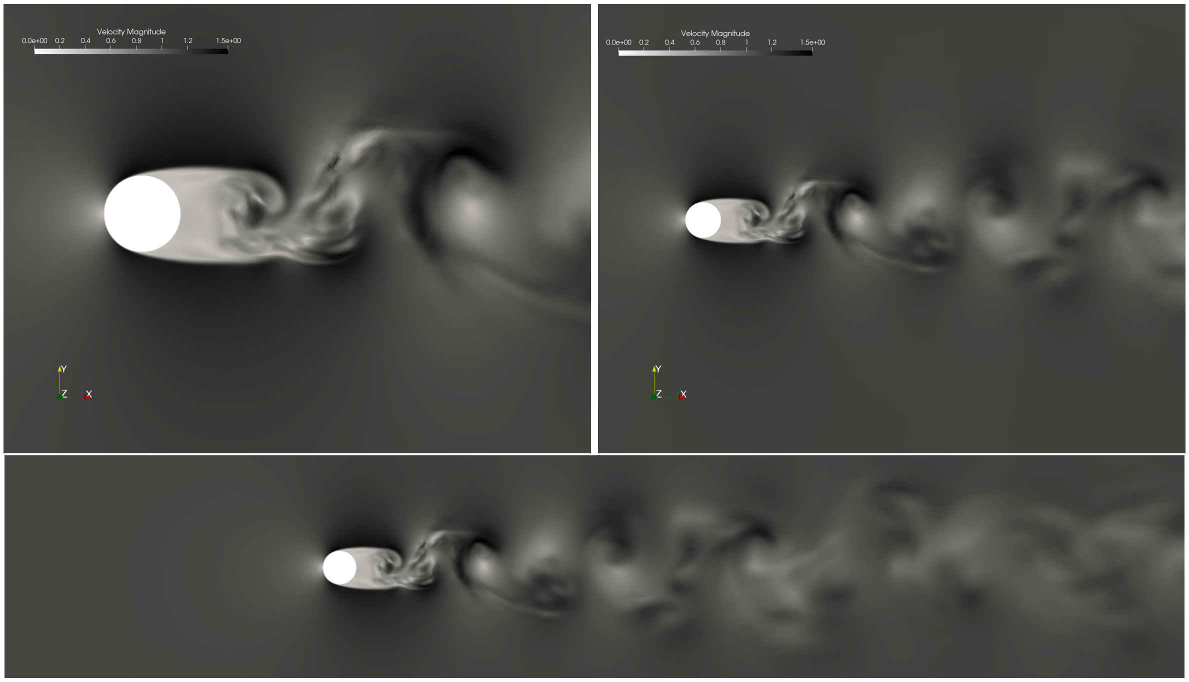


FIGURE 6.4 – Cylinder $Re = 3900$ VMS-WALE mesh adaptive simulation results : Velocity field in cross-section (on final mesh).

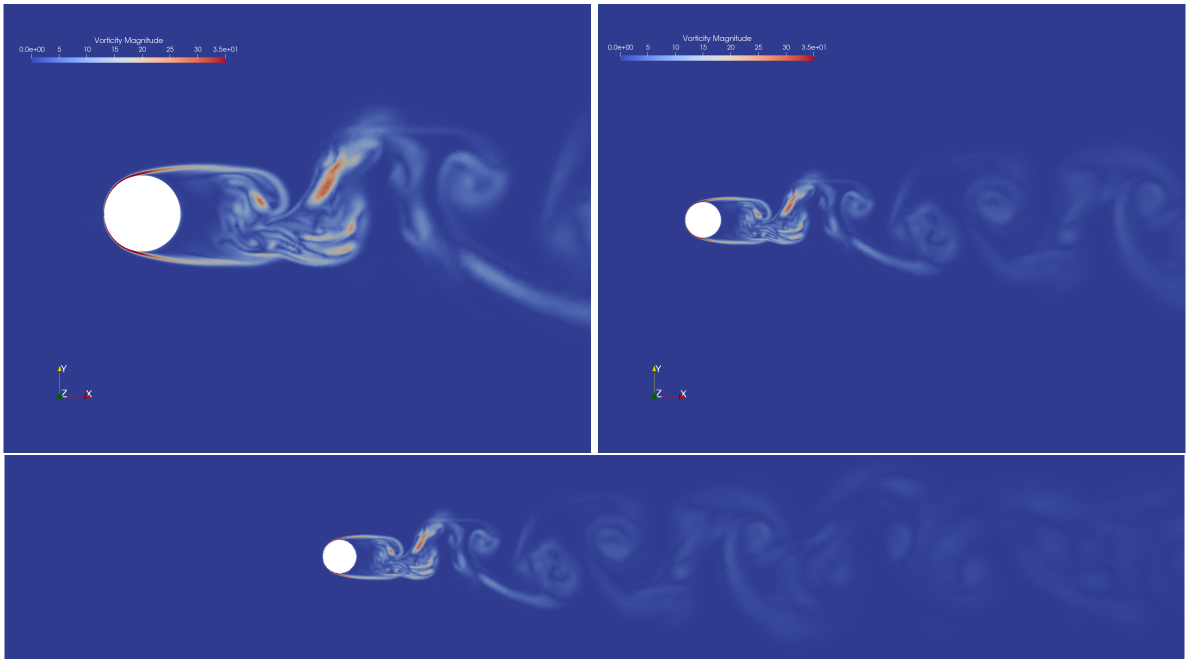


FIGURE 6.5 – Cylinder $Re = 3900$ VMS-WALE mesh adaptive simulation results : Vorticity field in cross-section (on final mesh).

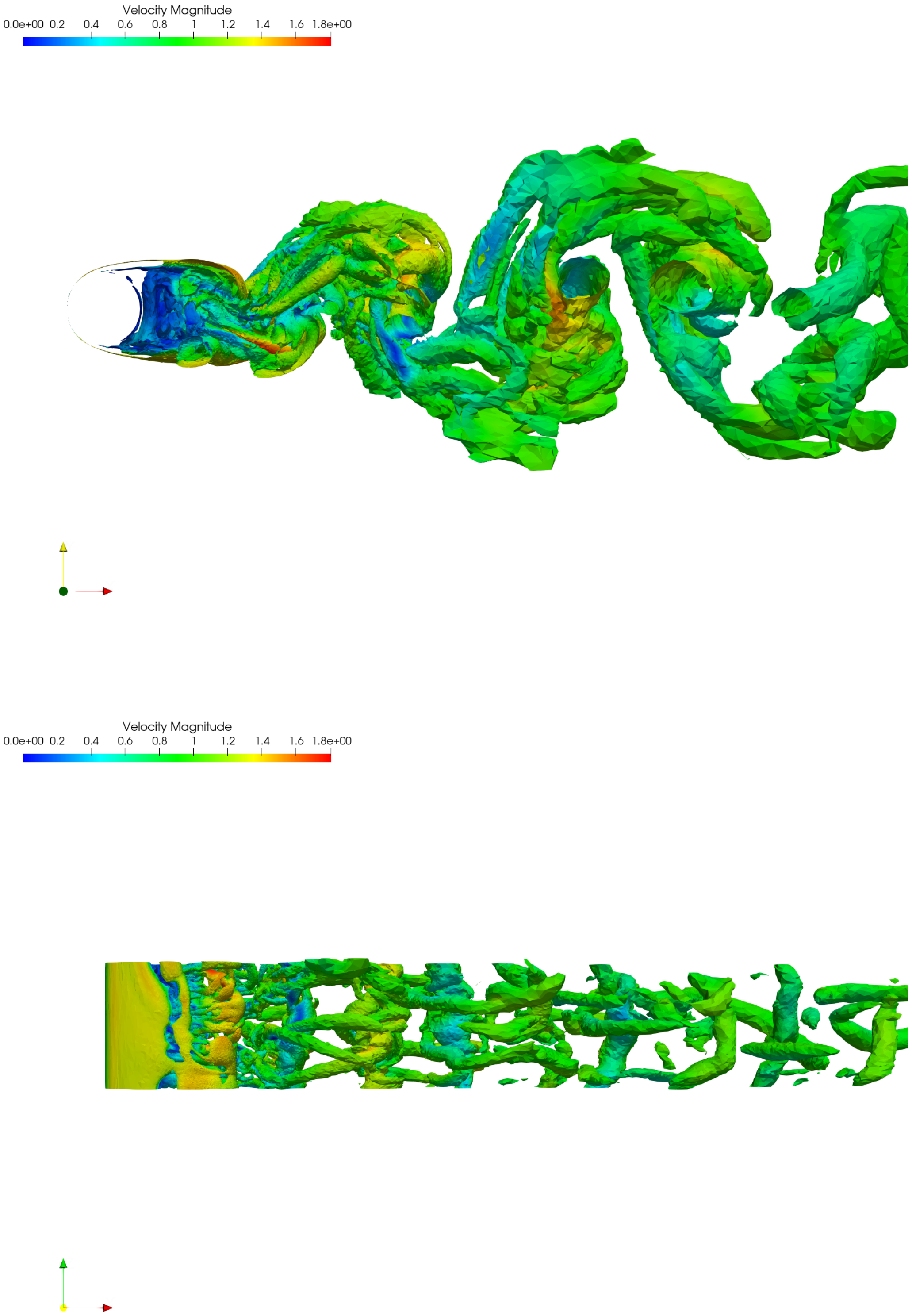


Figure 6.6 – Cylinder $Re = 3900$ VMS-WALE mesh adaptive simulation results : QCriterion (on final mesh).

4.2 VMS-WALE, CENO

The second numerical innovation of this thesis is the developement and test of the CENO3 approximation of advection terms described in Chapter 5. The computation has been performed with the same mesh of 1.4M vertices as the other non-adaptive calculations. The bulk outputs (Table 6.1) present unfortunatly a slighly less good accordance with the measurements.

4.3 3900 Synthesis

Let us comment Table 6.1. The first line is obtained by using with adequate options the computer code AiroNun used for many similar studies [58, 113, 94]. In order to fix some experimental values, we choose $\overline{C}_d \approx .99, C_l' \approx .09, l_r \approx 1.47$, -$\overline{C}_{pb} \approx .88, St \approx .21$. Not astonishingly, the presented results recomputed recently are in a rather good accordance with experiments with respectively relative deviations of 1% for $\overline{C}_d$, 1% for l_r, 1% for -$\overline{C}_{pb}$, 0.5% for St, but 54% for C_l'. The poor performance in C_l' prediction desserves some discussion. Numerically, we are looking to the convergence of a pressure time derivative, more difficult to obtain than other bulk quantities not involving derivation. As modelling is concerns, it is probable that LES filtering influences much the dynamics of the numerical flow, and therefore the prediction of lift fluctuations. Experimentally, C_l' is difficult to obtain. Only Norberg in [100] proposes measures for Reynolds numbers of 1500 and 4400. We have linearly interpolated between these measures and obtained the value of 0.0905.

In lines 2,3,4,5, the effect of using in the same code a RANS ingredient close to the boundary increases the disagreement nearly by an order of magnitude. Hybrid formulations show some improvement. Note also that the introduction of the transition modeling tends to improve notably the prediction of these RANS-based formulations.

In line 6, entitled NF VmsWale, we have run a commercial software with same options as in line 1 (and same mesh). Unfortunately we had a too low control of the details of approximation (numerical diffusion, control of timestep lenghth). Some outputs are rather well predicted, some other ones poorly predicted.

This second software involves a fourth-order spatial integration which, when used of the same mesh did not improved the prediction accuracy (line 7).

As already mentioned in Section 4.1, the mesh adaptive run gives much more acceptable results with a good pressure field (compared with experiments in Figure 6.3) obtained with a mesh with 50% less vertices, line 8.

	Mesh	$\overline{C}_d$	C'_l	l_r	-$\overline{C}_{pb}$	St	$\bar{\theta}$
Present adaptive simulation							
NF VmsWale	**adap**	**1.02**	**0.09**	**1.48**	**0.85**	**0.20**	**90**
Present simulations							
AN VmsWale	M1	1.00	0.17	1.42	0.88	0.211	94
AN k-ε	M1	0.91	0.17	1.24	0.84	0.22	
AN k-ε T	M1	0.91	0.16	1.25	0.84	0.22	
AN Hybrid	M1	0.89	0.13	1.38	0.80	0.22	
AN Hybrid-T	M1	0.90	0.14	1.36	0.81	0.22	
NF VmsWale	M1	1.02	0.12	1.88	1.01	0.192	
NF-Ceno VmsWale	M1	0.91	0.12		0.98	0.20	
Other simulations							
LES [104, 1, 68, 108]		[.99-1.38]		[1.0-1.56]	[.89-1.23]	[.19-0.21]	
Experiments							
Dong [112]				1.47			
Ong [69]						.210 ± .05	
Parnaudeau [104]				1.51 ± .15		.208 ± .002	
Lourenço [86]				1.18 ± .05			
Cardell [30]				1.33		.215	
Norberg from [1]		.99 ± .05	.09		.88 ± .005		
Kravchenko[1]		.99 ± .05		1.49 ± .02	.87 ± .05	.21 ± .1	

TABLE 6.1 – Bulk coefficient of the flow around a circular cylinder at Reynolds number $\mathbf{3900}$. $\overline{C}_d$ holds for the mean drag coefficient, C'_l is the root mean square of lift time fluctuation, $\overline{C}_{p_b}$ is the mean pressure coefficient at cylinder basis. *Models :* $T = k - \varepsilon - \gamma$, Vms = Variational multi-scale, Wale = Wall Adapted Large Eddy Simulation [98], Hybrid = $k - \varepsilon$ + VmsWale, Hybrid-T= $k - \varepsilon - \gamma$ + VmsWale. *Numerics :* AN=AiroNum, MUSCL-V6, NF=NiceFlow MUSCL-V6, NF-Ceno=NiceFlow, 4th-order Ceno. *Meshes :* adapt = adaptive, 903KNodes, M1 = non-adapted medium mesh 1.4MNodes. Experimental C'_l= RMS of the C_{pbase} pressure divided by $\bar{C}_{pbase}$.

5 Subcritical 20K

The $Re = 20K$ case is similar to the 3900 case (laminar boundary layer, turbulent wak) but involves a more complex vortex system.

5.1 VMS-WALE Adaptatif

We choose again the Transient Fixed Point adaptation algorithm and the Hessian of Mach as anisotropic criterion. The final mesh together with velocity and vorticity fields are depicted in Figure 6.7, where the mesh has $1.1M$ vertices.

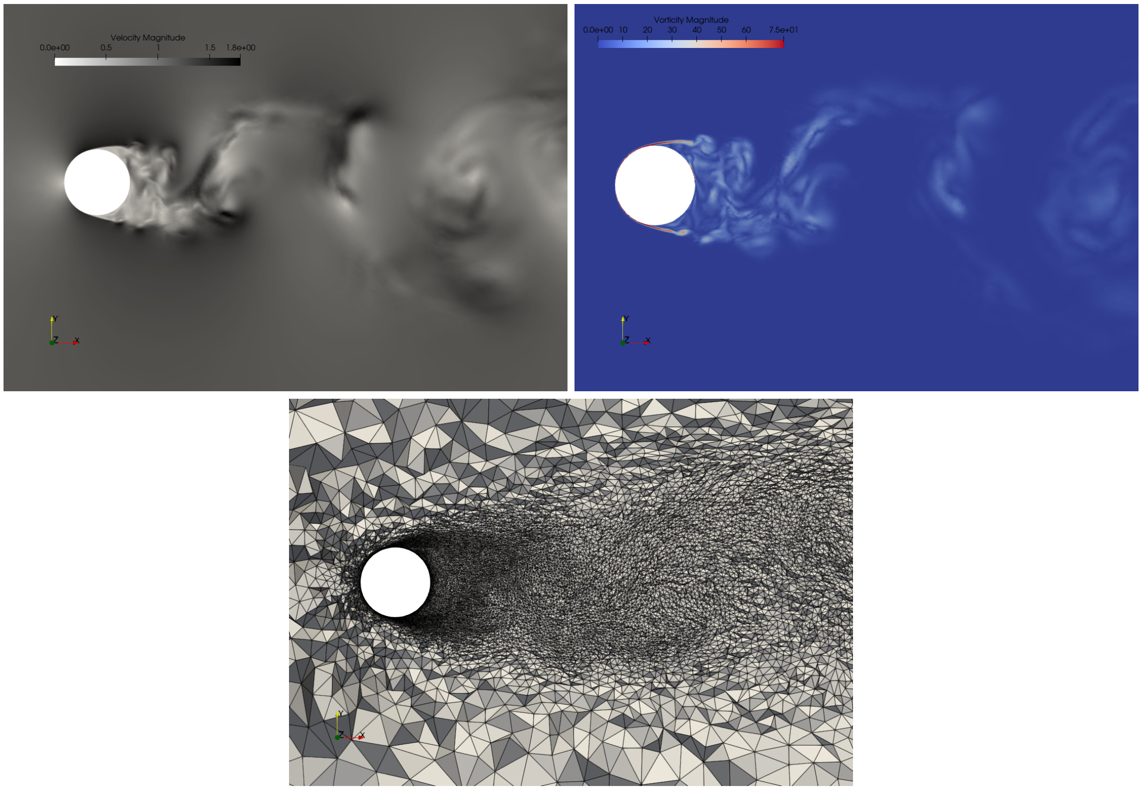


FIGURE 6.7 – Cylinder $Re = 20K$ VMS-WALE mesh adaptive simulation results : (bottom) final mesh in cross-section, (top) velocity field and vorticity field in cross-section (on the final mesh).

As for Reynolds 3900, the mean C_p distribution is in good accordance with the measurements, Figure 6.8.

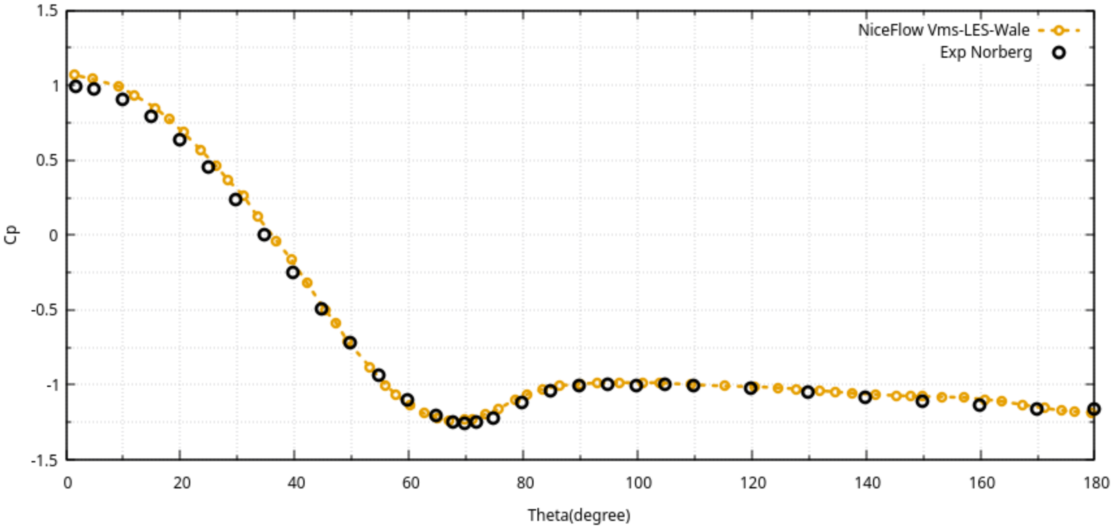


FIGURE 6.8 – Cylinder $Re = 20K$ VMS-WALE mesh adaptive simulation results : distribution over the cylinder surface of the mean pressure coeficient compared with the experimental measures of Norberg [100].

The complexity of the vortices system is not captured with the 1.1M vertices adaptive calculation and we depict in Figure 6.9 an adaptive calculation with higher complexity, producing and adapted mesh of 3.1M vertices. The Q criterion is also depicted in Figure 6.10. The Mach number adaptation criterion allows a rather fine discretization on the wake region at less than 5-6 chords of the cylinder. Bulk coefficients are rather well predicted, see Table 6.2.

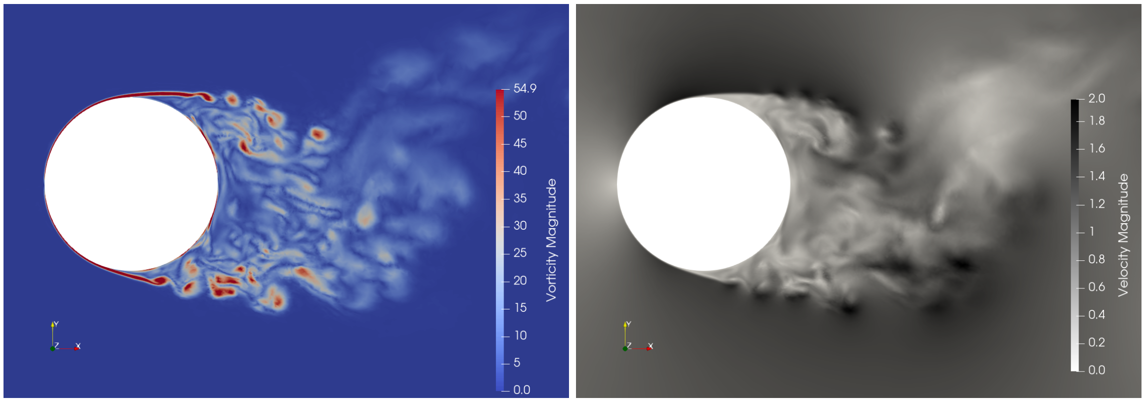


FIGURE 6.9 – Cylinder $Re = 20K$ VMS-WALE mesh adaptive simulation results : Vorticity field (right) and velocity field (left) in cross-section, for a finer adapted mesh, of the order of 3.1M vertices.

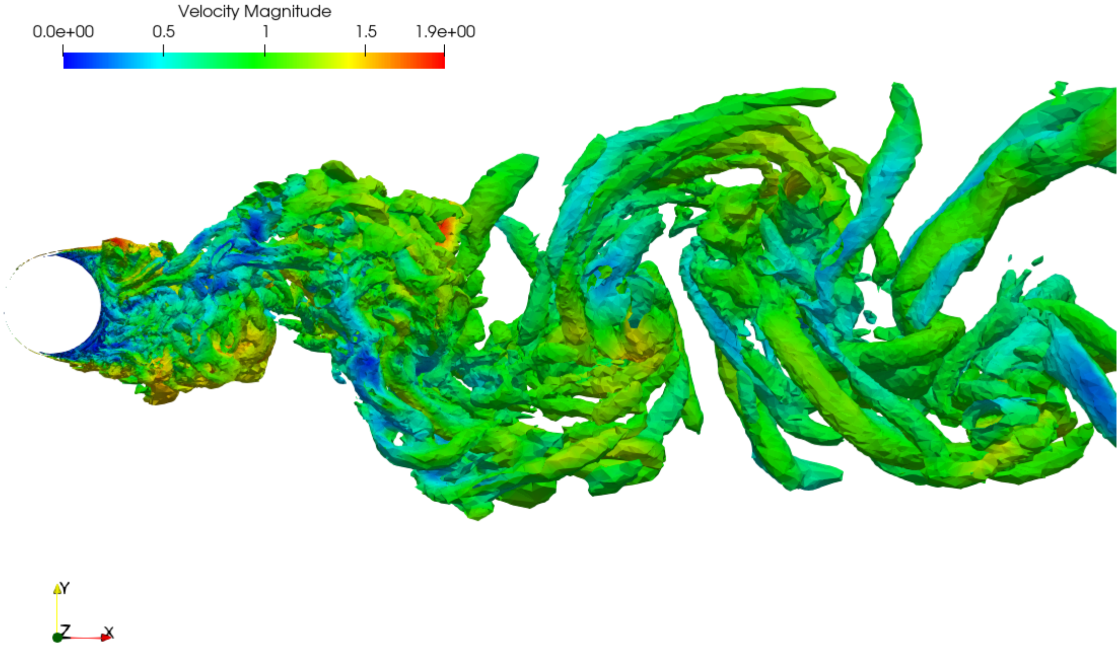


FIGURE 6.10 – Cylinder $Re = 20K$ VMS-WALE mesh adaptive simulation results : QCriterion (on final mesh).

5.2 Synthesis

For this test case, we have less measurements. There is an uncertainty for the C_{p-base} which is around $1. - 1.1$ in the cited experiments and 1.19 in other compilations.

We observe that NF VmsWale-adap, AN $k-\varepsilon-\gamma$/DVMS, and a less recent computation with AN VmsWale are in a first set of methods with prediction in a rather good agreement with experiments, namely 2-4% for C_d, 6-10% for C'_L, 10% for C_{pbase}, 8% for the Strouhal. In particular, AN $k - \varepsilon - \gamma$/DVMS seems to behave as well as the LES calculations. Same remark for the mesh adaptive computation, that performs well compared to other numerical and experimental results .

As already remarked for Reynolds 3900, AN $k - \varepsilon$/DVMS is less good than the three above : 10% for C_d, 20% for C'_L, 20% for C_{pbase}, 12% for Strouhal.

	Mesh	$\overline{C}_d$	C'_l	-$\overline{C}_{pb}$	St
Present simulation adaptive					
NF VmsWale	**adap**	**1.24**	**0.45**	**1.08**	**0.193**
Present simulation					
AN k-ε/DVMS	M2	1.102	0.60	0.85	0.22
AN k-ε-γ/DVMS	M2	1.227	0.48	1.19	0.21
Other simul.					
LES Aradag[15]				$[1.04 - 1.25]$	
LES Salvatici[115]		$[0.94 - 1.28]$	$[0.17 - 1.65]$	$[0.83 - 1.38]$	
AN VmsWale [63]		1.17	.42	1.20	.20
Experiments					
Norberg [100]			0.47		0.194
Lim [79]		1.2		1.08	
Yokuda [139]				1.	
Basu [21]		1.17			

Table 6.2 – Bulk coefficients of the flow around a circular cylinder at Reynolds number $20K$. $\overline{C}_d$ holds for the mean drag coefficient, C'_l is the root mean square of lift time fluctuation, $\overline{C}_{p_b}$ is the mean pressure coefficient at cylinder basis. *Models :* Hybrid = $k - \varepsilon - \gamma$ + SGS model, trans = $k - \varepsilon - \gamma$, *Numerics :* NF=NiceFlow, AN=AiroNum. *Meshes :* adapt = adaptive, 1.1MNodes, M2 = non-adapted medium mesh 1.8MNodes.

6 Supercritical 1M

This time, we reach the supercritical regime, with a Reynolds number of $1M$. In this regime, the boundary layer is laminar and turbulent. The laminar-turbulent transition is located between the stagnation and detachment points.

6.1 DDES-WL (wall law) Adaptive

The Transient Fixed Point adaptation algorithm is still applied and the Hessian of Mach as anisotropic criterion. The final mesh together with velocity and vorticity fields are depicted in Figure 6.12, where the mesh has $1.3M$ vertices.

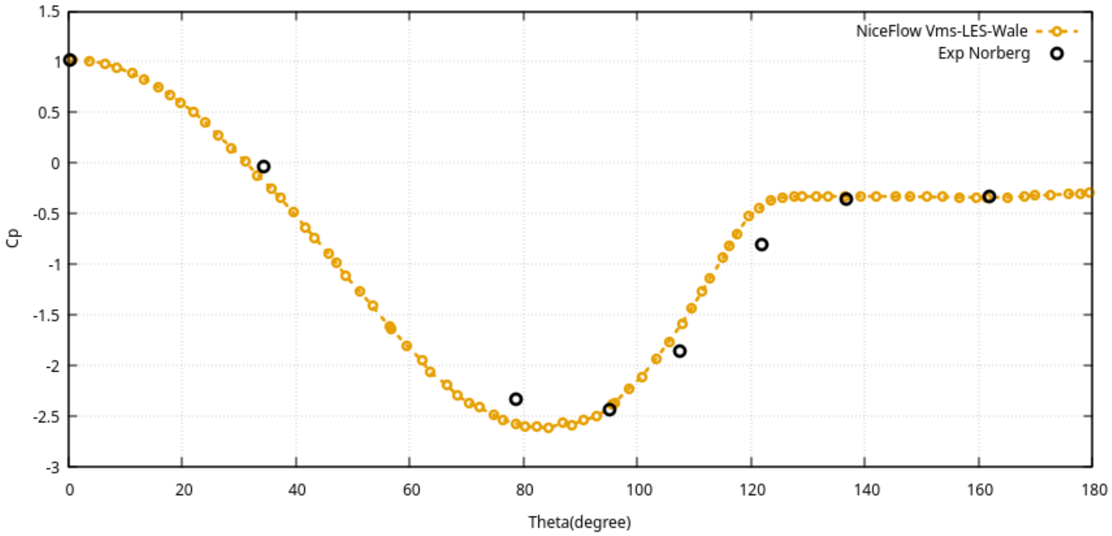


FIGURE 6.11 – Cylinder $Re = 1M$ DDES (wall law) mesh adaptive simulation results : distribution over the cylinder surface of the mean pressure coeficient compared with the experimental measures of Norberg [100].

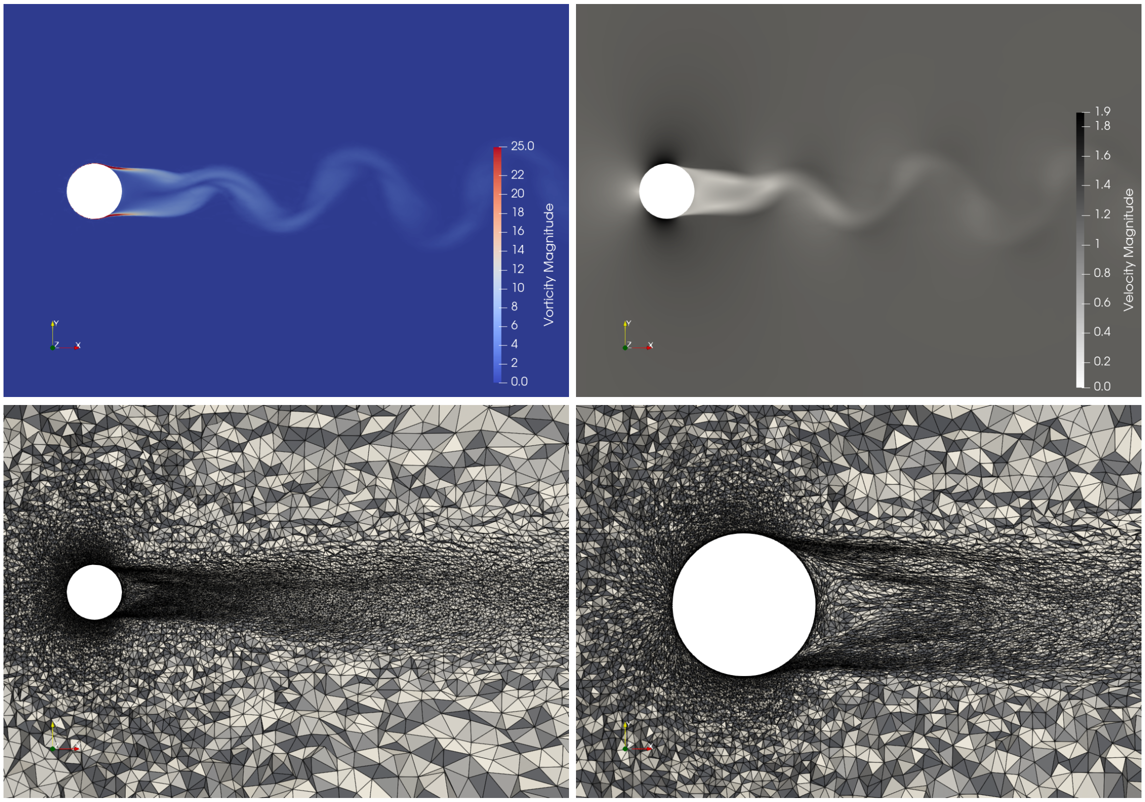


FIGURE 6.12 – Cylinder $Re = 1M$ DDES (wall law) mesh adaptive simulation results : (bottom) final mesh in cross-section, (top) velocity field and vorticity field in cross-section (on the final mesh).

6.2 Supercritical 1M, synthesis

The mesh-adaptive calculation NF DDES-WL and the transition computation AN $k-\varepsilon-\gamma$/DVMS give rather good predictions for the mean quantities $\overline{C}_d$, $|\overline{C_L}|$, not too bad for C_l' but the mesh-adaptive NF DDES-WL gives a Strouhal of 0.36,close to RANS-based calculations, while the transition AN $k-\varepsilon-\gamma$/DVMS proposes the maximal Strouhal 0.50, one of the two Strouhal identified by the experiments.

	Mesh	$\overline{C}_d$	$\lvert\overline{C_L}\rvert$	C_l'	$-\overline{C}_{pb}$	θ_{sep}	St
Present simulation adaptive Re=10^6							
NF DDES-WL	**adap**	**0.249**	**0.009**	**0.014**	**0.29**	**124**	**0.36**
Present simulation Re=10^6							
AN $k-\varepsilon$/DVMS		0.536	0.03	0.30	0.51	110	0.34
AN $k-\varepsilon-\gamma$/DVMS		0.289	0.03	0.04	0.25	128	0.50
Simulation							
LES of Kim and Mohan [71]		0.27	-	0.12	0.28	108	-
LES of Catalano et al. [103]		0.31	-	-	0.32	-	0.35
Experiments							
Shih [120]		0.24	-	-	0.33	-	-
Schewe [117]		0.22	-	0.02	-	-	0.44
Gölling [54]		0.22	-	-	-	130	0.12/0.47
Zdravkovich [140]		0.2-0.4	-	0.1-0.15	0.2-0.34	-	0.18/0.50
Roshko [111]		0.29	-	-	0.34	-	-

Table 6.3 – Bulk coefficients of the flow around a circular cylinder at Reynolds numbers 7.2×10^5 and 10^6 (super-critical regime). Same symbols as in Table 6.1

7 Concluding remarks

Turbulenct CFD relying on LES and hybrid models is a quite CPU intensive tool. With the same mesh as for RANS models, the computational cost is at least one order of magnitude higher due to the long time interval to be computed before final statistics are pertinent. Now, in practice, the same mesh as for good RANS calculations is generally not sufficiently fine, which pushes the extra cost towards several orders of magnitude. In many identified CFD configurations, the LES or hybrid calculations give a useable prediction while RANS does not. But even in this favourable situation, the absolute accuracy of the prediction is still very limited. In the case of a flow past a single circular cylinder, best output are still far by several %'s from the measurements, most outputs are far by tens %'s. Trying much finer meshes, in order to enjoy the "LES convergence", is frequently disappointing and difficult to apply in the industrial context, but it remains a good direction of investigations for research.

In the present work, we use relatively coarse meshes but try to evaluate the possible improvements coming from different parts of the global method :

- models,
- numerics.

The *transition model* developed during the thesis of Florian Miralles carries a real improvement in the quality of prediction, in the whole range of Reynolds number.

The use of a *higher order accurate approximation* brings an inegal contribution. This seems to indicate that the source of inaccuracy is in some case more due to numerics, in some case more due to modeling. This is a pathway which needs further experiments in order to identify for which type of turbulent simulation higher order truncature can produce better predictions.

This thesis also permitted to elaborate *anisotropic mesh adaptive strategies* for a first families of turbulent LES/hybrid flows. The efficiency and accuracy of the steady RANS mesh adaptive case was already well established in a series of papers and in the book [39]. For LES and hybrid simulations of vortex shedding flows like those considered in this Chapter, the work in this thesis has identified a first mesh adaptation strategy. We keep the feature-based Mach-number criterion. The flow is now an unsteady one. We use the Transient fixed point mesh adaptation algorithm, but with only on mesh for the whole time interval (it is changed by the fixed point). This chapter validates our strategy while allowing to identify the necessary further improvements :

- since LES/hybrid calculations are advanced with an implicit time scheme, the timestep length must be controlled by the mesh adaptation loop in order to maximize the effect of adaptation in the accuracy/efficiency tradoff. This issue will be the object of Chapter 8.

- since LES/hybrid vortices are not well detected by the Mach-number criterion, we have to derive an analysis of the LES error. This question will be the object of Chapter 9.

CHAPITRE 7

Rotating machine and mesh adaptation

1 Introduction

An important issue in the numerical study of stator/rotor CFD configurations is the coupling of stator with rotor. A part of the contribution in the Norma consortium for this subject concerns (1) an *immersed boundary method* in which only one mesh is used, but the rotation of a part of the boundary is performed by a particular intersection of a part of the geometry boundary with the background mesh, (2) *sliding methods* in which the two domains (around stator and around rotor) are coupled through an interface along with the rotor domain is sliding. We refer to the common paper [3] for a description of the methods developed and for numerial experiments. In this thesis, we have contributed partly to the development and to the study of two other methods, namely a Chimera method and a Multiple Reference Frame method, and in particular to this combination with a mesh adaptation approach.

2 A first approach : The Chimera method

2.1 Theoretical aspect

The purpose is to solve the Navier-Stokes equations, which are recalled in condensed form :

$$\left\{ \begin{array}{ll} \partial_t W(t,\mathbf{x}) + \mathrm{div} F^C(W(t,\mathbf{x})) = \mathrm{div} F^D(W(t,\mathbf{x})) & (t,\mathbf{x}) \in [0,+\infty[\times\Omega, \\ W(t,\mathbf{x}) = \Phi_B(t,\mathbf{x}) & (t,\mathbf{x}) \in [0,+\infty[\times\partial\Omega, \end{array} \right. \tag{7.1}$$

where we divide Ω into two subdomains Ω_1 and Ω_2, such that $\Omega_1 \cap \Omega_2 \neq \emptyset$ and with $\Omega_2 \subset \Omega_1$. The aim of the decomposition method lies in recovering the global solution W of problem (7.1) from W_1 and W_2, respectively the solutions in the Ω_1 and Ω_2 domains. This involves describing what happens in $\Omega_1 \cap \Omega_2$. We will do this in the particular case of the Chimera method.

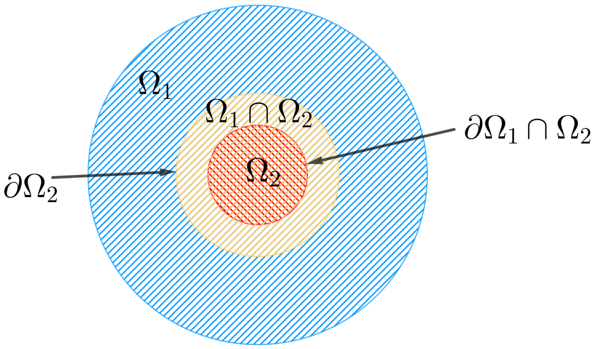


FIGURE 7.1 – Example of domain decomposition. In blue the Ω_1 domain, red the Ω_2 domain and yellow the intersection of the two domains.

For the standard Chimera method we consider the tetrahedrisation $\mathcal{T}_h$ of $\Omega_h \subset \Omega$. That is, Ω_h is decomposed into two subdomains Ω_h^1 and Ω_h^2 such that $\Omega_h^2 \subset \Omega_h^1$, and we impose that $\Omega_h^1 \cap \Omega_h^2 \neq \emptyset$. In this method, the points of the largest mesh (Ω_h^1 in our case) are deactivated in $\Omega_h^1 \cap \Omega_h^2$ and values at these points are recovered by interpolation in the element of Ω_h^2. The boundary values of the Ω_h^2 domain are also found by interpolating the values on the elements in Ω_h^1. Let us keep denoting φ_i our basis functions associated with the point $\mathbf{x}_i$ and N_{K_i} the set of vertices of the element K in which $\mathbf{x}_i$ is included ($\mathbf{x}_i$ is not a vertex of K_i). In terms of equations, the Chimera subproblems can be written as follows

$$\begin{cases} \partial_t W_2(t^{n+1}, \mathbf{x}_i) + \mathrm{div} F^C(W_2(t^{n+1}, \mathbf{x}_i)) = \mathrm{div} F^D(t^{n+1}, W_2(\mathbf{x}_i)) & \mathbf{x}_i \in \Omega_h^2, \\ W_2(t^{n+1}, \mathbf{x}_i) = \sum_{\mathbf{x}_k \in K_i} W_1(t^n, \mathbf{x}_k)\varphi_k(\mathbf{x_i}) & \mathbf{x}_i \in \partial\Omega_h^2, \end{cases}$$

$$\begin{cases} \partial_t W_1(t^{n+1}, \mathbf{x}_i) + \mathrm{div} F^C(W_1(t^{n+1}, \mathbf{x}_i)) = \mathrm{div} F^D(W_1(t^{n+1}, \mathbf{x}_i)) & \mathbf{x}_i \in \Omega_h^1, \\ W_1(t^{n+1}, \mathbf{x}_i) = \Phi_B(t^{n+1}, \mathbf{x}_i) & \mathbf{x}_i \in \partial\Omega_h^1 \backslash \partial\Omega_h^1 \cap \partial\Omega_h^2, \\ W_1(t^{n+1}, \mathbf{x}_i) = \sum_{\mathbf{x}_k \in K_i} W_2(t^{n+1}, \mathbf{x}_k)\varphi_k(\mathbf{x_i}) & \mathbf{x}_i \in \Omega_h^1 \cap \Omega_h^2. \end{cases}$$

2.2 Adaptation of the Chimera method to rotor/stator

We describe a first approach for rotor/stator interaction. The Chimera method aims at solving partial differential equations by decomposition into subdomains with overlap in order to avoid having to use a global mesh. These domains can be fixed or mobile, but must have an overlapping area because this method allows the communication between the computational domains thanks to their overlapping. For our rotor/stator applications, we use the Chimera methods to decompose the computational domain $\Omega \subset \mathbb{R}^3$ into two subdomains, one fixed domain Ω_f (the stator part) and one rotating domain Ω_r (the rotor part), see Fig.7.2. Consider a tetrahedrization $\mathcal{T}_f = \{T_f\}$ of Ω_f and a tetrahedrization $\mathcal{T}_r = \{T_r\}$ of Ω_r. Let $\{p_i^f\}_{i=1,n}$ the vertices of $\mathcal{T}_f$ and $\{p_j^r\}_{j=1,m}$ the vertices of $\mathcal{T}_r$, a crucial assumption is that, in $\Omega_f \cap \Omega_r$, each node p^f is internal to a tetrahedron T_r and reciprocally, each node p^r is internal to a tetrahedron T_f. Note that the boundary nodes, those located in overlap area, are deactivated, the values of our unknows are found at these points by a P_1 interpolation.

We denote by W_f (respectively W_r) the solution computed by our scheme in the domain $\Omega_f \backslash \Omega_f \cap \Omega_r$ (respectively the solution computed by our scheme in the domain $\Omega_r \backslash \Omega_f \cap \Omega_r$). An iteration in time goes as follows

- The domain Ω_r rotates by an angle α with respect to the domain Ω_f.
- We locate the boundary nodes p^f in Ω_r, and reciprocally we locate the boundary p^r in Ω_f.
- The values of W_f and W_r in $\Omega_f \cap \Omega_r$ are determined by a P_1 interpolation.
- Finally W_f and W_r are computed respectively in $\Omega_f \backslash \Omega_f \cap \Omega_r$ and $\Omega_r \backslash \Omega_f \cap \Omega_r$ with the new interpolated values.

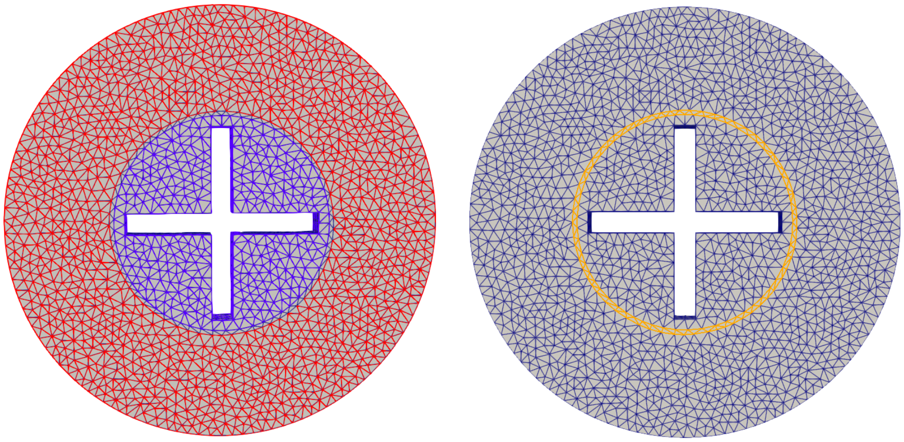


FIGURE 7.2 – Definition of rotor and stator from an initial mesh : in red, the stator, in blue the rotor, and in yellow, their common part.

At each time step the rotating domain will rotate with respect to the fixed domain according to the rotation speed we have imposed. For more clarity, let us zoom in on an area of the domain with overlap, we can see in Fig.7.3 the position of the meshes at the initial time and their position after one time step.

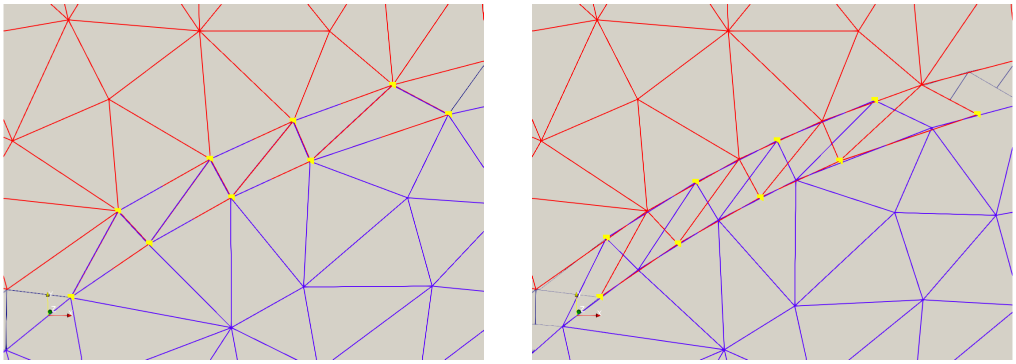


FIGURE 7.3 – Initial position of meshes, left, and the two meshes after a slight rotation of rotor.

The points in yellow correspond to points that are no longer in their respective mesh, no equation is solved on these points. We calculate the value of these points by interpolation. For example in Fig.7.4, the value of point 1 is found by interpolation in the blue triangle where it is inscribed.

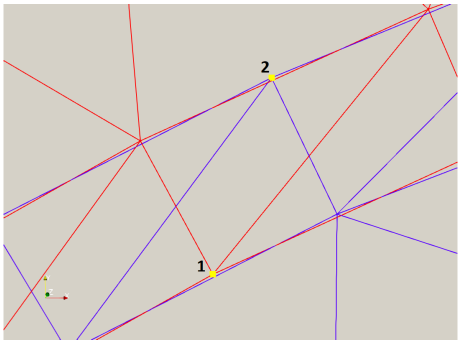


FIGURE 7.4 – Interpolation of hanging vertices (in yellow).

2.3 Introduction of mesh adaptation

2.3.1 Anisotropic mesh adaptation algorithm

We apply the Transient Fixed Point algorithm in combination with the Chimera solver. For a total time interval of 2 rotations, the total interval is divided into 16 subinterval in such a way that 16 different adapted meshes are used for the 2 rotations.

2.3.2 Example

For our example, the computation domain is the same as that used to schematize the Chimera method previously, in other words a disk of radius 2 meters and thickness 0.3

meter. A blade of the cross is 0.9 meter long and 0.2 meter wide.

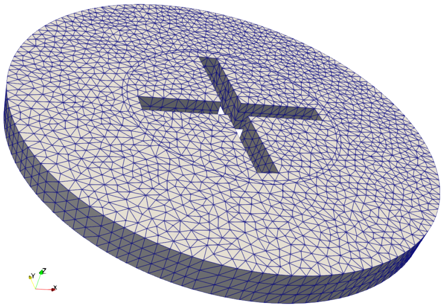


FIGURE 7.5 – 3D view of our domain.

The initial conditions of the following results are $\rho_0 = 1,2\,kg.m^{-3}$ in the stator part, $\rho_0 = 1,4\,kg.m^{-3}$ in the rotor part (see Fig.7.2), $\mathbf{v}_0 = 0$, $p_0 = 101325\,Pa$, $T_0 = 288,15\,K$ and the rotation speed is 1000 rpm. For the turbulence model we also initialize the Spalart variable at $10^{-6}\,m^2.s^{-1}$.

We briefly present two results, one obtained with mesh adaptation on the density and Mach number (Figure 7.6), and another with adaptation only on the Mach number (Figure 7.7).

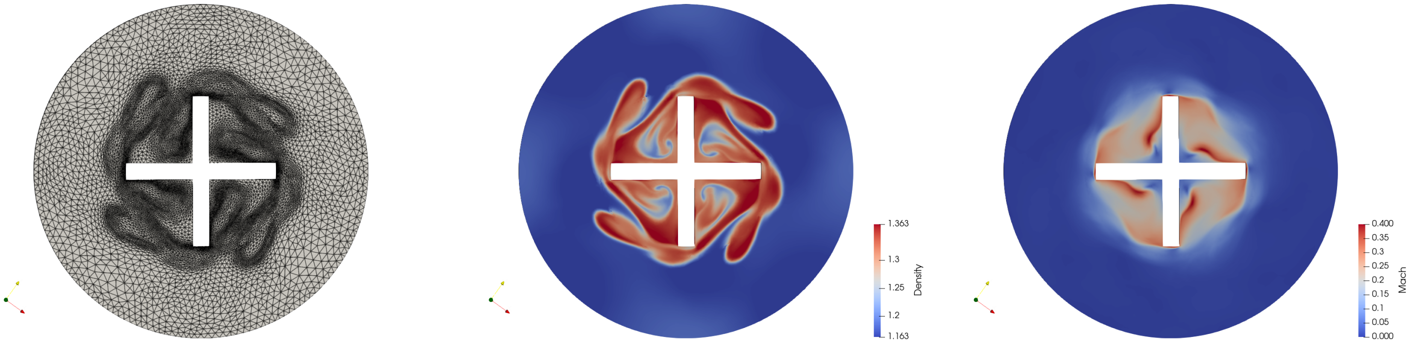


FIGURE 7.6 – Adapted mesh on the Mach number and the density at the left, the density at the middle and the Mach number on the right, after 5/8 turns of the cross.

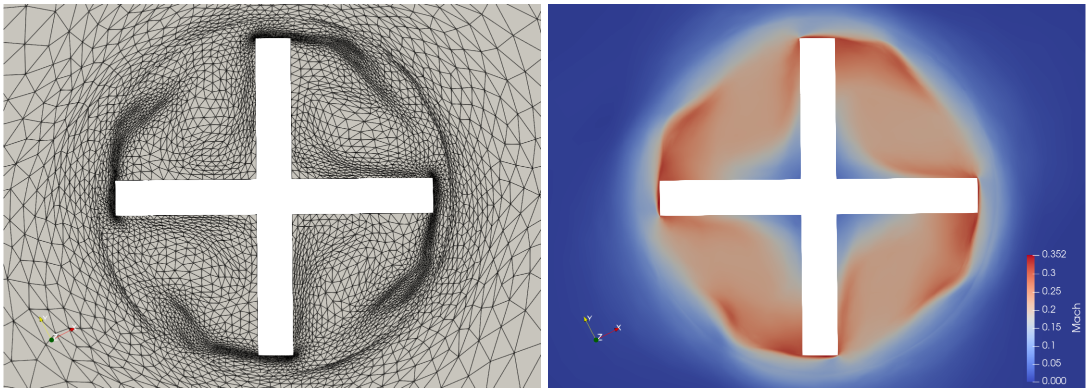


FIGURE 7.7 – Adapted mesh on the on the left and the Mach number on the right, after 2 rotation of the cross.

The mean number of vertices in the different meshes is $50K$. The stable CFL number is solely about 5. This means that the number of timesteps is large and the method rather computer cost consuming

This method will not be used for the rest of our study. The first reason is that this method is prohibitively expensive in terms of computation time, for cases more complex than the one presented above, the computation times become unreasonable.

2.3.3 Conclusion on Chimera

We have introduced successfully a combination of the Chimera approach and anisotropic mesh adaptation. The method is time consistent. The method is advanced in time implicitly, but cannot reach high enough CFL number for an efficient application to the stator-rotor configurations we are interested with. Therefore the application of this combination will not be applied to most of the quasi-steady calculation but will be restricted to particular cases where time consistency is mandatory.

3 A second approach : Multiple Reference Frame (MRF)

The method chosen to take into account the rotor rotation in our simulations consists in reformulating the Navier-Stokes equations in a rotating reference frame. We will describe two of these methods in this section, namely the Single Reference Frame (SRF) method and the Multiple Reference Frame (MRF) method.

3.1 The Navier-Stokes equations in rotating frame

We consider two reference frames, $\mathcal{R}$ and $\mathcal{R}'$, with $\mathcal{R}$ the Galilean reference frame and $\mathcal{R}'$ in rotation with respect to $\mathcal{R}$ around an axis $\mathbf{e}$, with constant angular velocity $\boldsymbol{\omega}$. Let $\mathbf{x}$ be a positive vector, the law of velocity compositions gives us

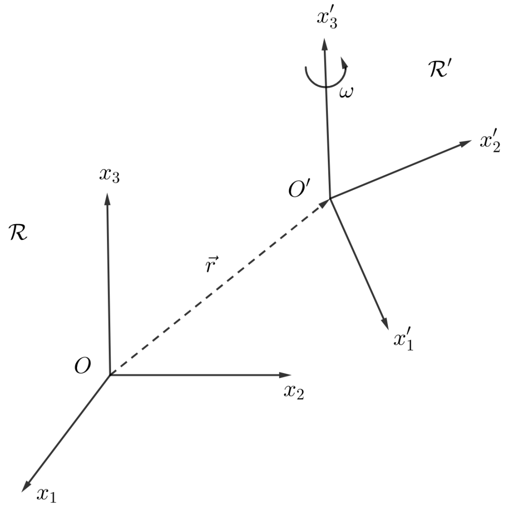


FIGURE 7.8 – Fixed and mobile references frames .

$$\left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\mathcal{R}} = \left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\mathcal{R}'} + \boldsymbol{\omega} \times \mathbf{x} \,.$$

Let us note the velocity vectors $\left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\mathcal{R}} = \mathbf{u}$ and $\left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\mathcal{R}'} = \mathbf{u}'$. We have

$$\begin{aligned}
\left.\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}\right|_{\mathcal{R}} &= \left.\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}\right|_{\mathcal{R}'} + \boldsymbol{\omega} \times \mathbf{u} \\
&= \left.\frac{\mathrm{d}(\mathbf{u}' + \boldsymbol{\omega} \times \mathbf{x})}{\mathrm{d}t}\right|_{\mathcal{R}'} + \boldsymbol{\omega} \times (\mathbf{u}' + \boldsymbol{\omega} \times \mathbf{x}) \\
&= \left.\frac{\mathrm{d}\mathbf{u}'}{\mathrm{d}t}\right|_{\mathcal{R}'} + 2\boldsymbol{\omega} \times \mathbf{u}' + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x}) \,.
\end{aligned}$$

Note that we have

$$\begin{aligned}\left.\frac{\mathrm{d}\,\mathbf{u}(t,\mathbf{x}(t))}{\mathrm{d}t}\right|_{\mathcal{R}} &= \frac{\partial \mathbf{u}}{\partial t} + \left(\left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\mathcal{R}} \cdot \nabla\right)\mathbf{u} \\ &= \frac{\partial \mathbf{u}}{\partial t} + \left(\mathbf{u}\cdot\nabla\right)\mathbf{u}\,.\end{aligned}$$

where $\mathbf{u}$ is the absolute velocity and $\mathbf{u}'$ is the relative velocity. In the presence of a fluid flow in a rotating frame, the velocity of the fluid can be expressed in the fixed frame and in the moving frame. The relation between the two velocities is $\mathbf{u}' = \mathbf{u} - \mathbf{u}_e$ with $\boldsymbol{u}_e = \boldsymbol{\omega} \times \mathbf{x}$. The compressible Navier-Stokes equations can be written in relative or absolute velocity, we will describe these two formulations.

3.1.1 Relative velocity formulation

In this subsection we will see how the equations of mass conservation, quantity of motion conservation and total energy are formulated in relative velocity.

• Mass conservation equation :

The mass conservation equation is

$$\frac{\partial \rho}{\partial t} + \mathrm{div}\,(\rho \mathbf{u}) = \frac{\partial \rho}{\partial t} + \mathrm{div}\,(\rho(\mathbf{u}' + \boldsymbol{\omega} \times \mathbf{x})) = \frac{\partial \rho}{\partial t} + \mathrm{div}\,(\rho \mathbf{u}') = 0.$$

• Quantity of motion conservation equation :

Starting from the accelerations composition, the motion equation is

$$\begin{aligned}\rho\left.\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}\right|_{\mathcal{R}} &= \rho\left.\frac{\mathrm{d}\mathbf{u}'}{\mathrm{d}t}\right|_{\mathcal{R}'} + \rho\left(\boldsymbol{\omega} \times \mathbf{u}' + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x})\right) = -\nabla p + \mathrm{div}\,\mathcal{T} \\ &= \rho\frac{\partial \mathbf{u}'}{\partial t} + \rho\left(\mathbf{u}' \cdot \nabla\right)\mathbf{u}' + \rho\left(2\boldsymbol{\omega} \times \mathbf{u}' + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x})\right) = -\nabla p + \mathrm{div}\,\mathcal{T},\end{aligned}$$

so considering the mass conservation equation and the motion equation we have

$$\begin{aligned}\frac{\partial(\rho \mathbf{u}')}{\partial t} &= \mathbf{u}'\frac{\partial \rho}{\partial t} + \rho\frac{\partial \mathbf{u}'}{\partial t} \\ &= -\mathbf{u}'\mathrm{div}\,(\rho\mathbf{u}') - \rho\left((\mathbf{u}' \cdot \nabla)\mathbf{u}' + 2\boldsymbol{\omega} \times \mathbf{u}' + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x})\right) - \nabla p + \mathrm{div}\,\mathcal{T}\end{aligned}$$

and as

$$\mathbf{u}'\mathrm{div}\,(\rho\mathbf{u}') + \rho(\mathbf{u}' \cdot \nabla)\mathbf{u}' = \sum_{i=1}^{3}\left(\mathbf{u}'\partial_i(\rho\mathbf{u}'_i) + \rho\mathbf{u}'_i\partial_i\mathbf{u}'\right) = \sum_{i=1}^{3}\partial_i\left(\rho\mathbf{u}'_i\mathbf{u}'\right) = \mathrm{div}\,(\rho\mathbf{u}' \otimes \mathbf{u}')\,,$$

finally

$$\frac{\partial(\rho \mathbf{u}')}{\partial t} + \operatorname{div}(\rho \mathbf{u}' \otimes \mathbf{u}') + \rho\left(2\boldsymbol{\omega} \times \mathbf{u}' + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x})\right) + \nabla p = \operatorname{div} \mathcal{T}\,.$$

• **Total energy conservation equation** :

Taking the mass conservation equation and what we did for the momentum conservation equation we have

$$\begin{aligned}\frac{\partial(\rho E_r)}{\partial t} &= \frac{\partial(\rho e)}{\partial t} + \frac{|\mathbf{u}'|^2}{2}\frac{\partial \rho}{\partial t} + \rho \mathbf{u}' \cdot \frac{\partial \mathbf{u}'}{\partial t} \\ &= -\operatorname{div}(\rho \mathbf{u}' e) - \frac{|\mathbf{u}'|^2}{2}\operatorname{div}(\rho \mathbf{u}') - \rho \mathbf{u}' \cdot ((\mathbf{u}' \cdot \nabla)\mathbf{u}') - \rho \mathbf{u}' \cdot (2\boldsymbol{\omega} \times \mathbf{u}' + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x})) \\ &\quad + \operatorname{div}(\lambda \nabla T + \mathcal{T} \cdot \mathbf{u}' - p\mathbf{u}')\,,\end{aligned}$$

we can easily verify that $\mathbf{u}' \cdot (\boldsymbol{\omega} \times \mathbf{u}') = 0$ and

$$|\mathbf{u}'|^2 \operatorname{div}(\rho \mathbf{u}') + \rho \mathbf{u}' \cdot ((\mathbf{u}' \cdot \nabla)\mathbf{u}') = \operatorname{div}\left(\rho \mathbf{u}' \frac{|\mathbf{u}'|^2}{2}\right).$$

So finally we have

$$\frac{\partial(\rho E_r)}{\partial t} + \operatorname{div}((\rho E_r + p)\,\mathbf{u}') + \rho \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x}) \cdot \mathbf{u}' = \operatorname{div}(\lambda \nabla T + \mathcal{T} \cdot \mathbf{u}')\,.$$

• **Relative velocity equations** :

To summarize, the Navier-Stokes equations in relative velocity, named as such because the unknown for the momentum conservation equation is $\rho\mathbf{u}'$, are given by :

$$\begin{cases} \partial_t \rho + \operatorname{div}(\rho \mathbf{u}') = 0, \\ \partial_t(\rho \mathbf{u}') + \operatorname{div}(\rho \mathbf{u}' \otimes \mathbf{u}') + \rho(\boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{x} + 2\boldsymbol{\omega} \times \mathbf{u}') + \nabla p = \operatorname{div} \mathcal{T}, \\ \partial_t(\rho E_r) + \operatorname{div}((\rho E_r + p)\mathbf{u}') + \rho \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x}) \cdot \mathbf{u}' = \operatorname{div}(\lambda \nabla T + \mathcal{T} \cdot \mathbf{u}'). \end{cases}$$

here $\mathcal{T} = \mu\left(\nabla \mathbf{u}' + {}^t\nabla \mathbf{u}' - \frac{2}{3}(\operatorname{div} \mathbf{u}')\mathbb{I}_3\right)$ and $E_r = e + \frac{1}{2}\|\mathbf{u}'\|^2$. The relative velocity formulation is appropriate when most of the fluid in the domain is in rotation. In this case, the method used is called Single Reference Frame (SRF).

But this method suffers from a major flaw, especially for the applications we are considering. It is impossible to define rotor/stator domains for propeller/fuselage calculations, for example, which makes this method of little interest for our studies. That's why we are going to take a look at a new method.

3.1.2 Absolute velocity formulation

The main change with respect to the relative velocity formulation is that this time we consider our motion equation in the following form

$$\begin{aligned}\rho\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}\bigg|_{\mathcal{R}} &= \rho\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}\bigg|_{\mathcal{R}'} + \rho\boldsymbol{\omega}\times\mathbf{u}\\ &= \rho\frac{\partial\mathbf{u}}{\partial t} + \rho\left(\mathbf{u}'\cdot\nabla\right)\mathbf{u} + \rho\boldsymbol{\omega}\times\mathbf{u} = -\nabla p + \operatorname{div}\mathcal{T},\end{aligned}$$

• Mass conservation equation :

The mass conservation equation is unchanged from the relative velocity formulation

$$\frac{\partial\rho}{\partial t} + \operatorname{div}\left(\rho\mathbf{u}'\right) = 0.$$

• Quantity of motion conservation equation :

Considering our mass conservation and motion equation we have

$$\begin{aligned}\frac{\partial(\rho\mathbf{u})}{\partial t} &= \mathbf{u}\frac{\partial\rho}{\partial t} + \rho\frac{\partial\mathbf{u}}{\partial t}\\ &= -\mathbf{u}\operatorname{div}\left(\rho\mathbf{u}'\right) - \rho\left(\mathbf{u}'\cdot\nabla\right)\mathbf{u} - \rho\boldsymbol{\omega}\times\mathbf{u} - \nabla p + \operatorname{div}\mathcal{T},\end{aligned}$$

we check that

$$\mathbf{u}\operatorname{div}\left(\rho\mathbf{u}'\right) + \rho\left(\mathbf{u}'\cdot\nabla\right)\mathbf{u} = \sum_{i=1}^{3}\left(\mathbf{u}\partial_i(\rho\mathbf{u}'_i) + \rho\mathbf{u}'_i\partial_i\mathbf{u}\right) = \sum_{i=1}^{3}\partial_i(\rho\mathbf{u}'_i\mathbf{u}) = \operatorname{div}\left(\rho\mathbf{u}'\otimes\mathbf{u}\right),$$

then we have

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \operatorname{div}\left(\rho\mathbf{u}'\otimes\mathbf{u}\right) + \rho\boldsymbol{\omega}\times\mathbf{u} + \nabla p = \operatorname{div}\mathcal{T}.$$

• Energy conservation equation :

$$\begin{aligned}\frac{\partial(\rho E)}{\partial t} &= \frac{\partial(\rho e)}{\partial t} + \frac{|\mathbf{u}|^2}{2}\frac{\partial\rho}{\partial t} + \rho\mathbf{u}\cdot\frac{\partial\mathbf{u}}{\partial t}\\ &= -\operatorname{div}\left(\rho\mathbf{u}'e\right) - \frac{|\mathbf{u}|^2}{2}\operatorname{div}\left(\rho\mathbf{u}'\right) - \rho\mathbf{u}\cdot((\mathbf{u}'\cdot\nabla)\mathbf{u}) - \rho\mathbf{u}\cdot(\boldsymbol{\omega}\times\mathbf{u}) + \operatorname{div}\left(\lambda\nabla T + \mathcal{T}\cdot\mathbf{u} - p\mathbf{u}\right)\end{aligned}$$

we can check again that $\mathbf{u}\cdot(\boldsymbol{\omega}\times\mathbf{u}) = 0$ and

$$\frac{|\mathbf{u}|^2}{2}\operatorname{div}\left(\rho\mathbf{u}'\right) + \rho\mathbf{u}\cdot((\mathbf{u}'\cdot\nabla)\mathbf{u}) = \operatorname{div}\left(\rho\mathbf{u}'\frac{|\mathbf{u}|^2}{2}\right).$$

Finally

$$\frac{\partial(\rho E)}{\partial t} + \operatorname{div}\left(\rho E \mathbf{u}' + p\mathbf{u}\right) = \operatorname{div}\left(\lambda \nabla T + \mathcal{T} \cdot \mathbf{u}\right).$$

• Absolute velocity equations :

$$\begin{cases} \partial_t \rho + \operatorname{div}(\rho \mathbf{u}') = 0, \\ \partial_t(\rho \mathbf{u}) + \operatorname{div}(\rho \mathbf{u}' \otimes \mathbf{u}) + \rho(\boldsymbol{\omega} \times \mathbf{u}) + \nabla p = \operatorname{div} \mathcal{T}, \\ \partial_t(\rho E) + \operatorname{div}((\rho E + p)\mathbf{u}' + p\,\mathbf{u}_e) = \operatorname{div}(\lambda \nabla T + \mathcal{T} \cdot \mathbf{u}). \end{cases}$$

The absolute velocity formulation is preferred in applications where the fluid flow does not rotate through most of the domain. In this case, the method used is called Multiple Reference Frame (MRF).

Some problems involve multiple moving parts or contain surfaces that are not surfaces of revolution. For these problems, it is appropriate to divide the computational domain into several fluid/solid cell zones, with interface boundaries separating the zones. The zones that contain the moving components can then be solved using the equations of the moving frame while the stationary zones can be solved with the equations of the fixed frame.

When the absolute velocity formulation is used, the equations in each domain are expressed in absolute velocity, so no transformation is required at the interface of the two domains in this approach.

3.2 Discretization of the absolute velocity equations

Now we are interested in the spatial discretization of the Navier Stokes system in absolute formulation. The description of the discretisation is the same as in the chapter on Numerical Methods, so we will only give the broad outlines. Let $\Omega_h \subseteq \Omega \subset \mathbb{R}^3$ a polygonal domain approximating our computational domain Ω and $\mathcal{T}_h = \{K\}$ its conformal tetrahedrization. Again the adopted scheme is vertex-centered and at each node i of the mesh we build a dual cell C_i such that

$$\Omega_h = \bigcup_{i=1}^{\mathcal{N}} C_i\,, \quad \mathcal{N} \text{ the number of nodes.}$$

We consider the Navier-Stokes system in absolute velocity formulation, written in condensed form,

$$\begin{cases} \partial_t W + \operatorname{div} F^C(W) = \operatorname{div} F^D(W) + F^S(W), & (t, \mathbf{x}) \in [0, +\infty[\times \Omega_h, \\ W(0, \mathbf{x}) = W_0(\mathbf{x}), & \mathbf{x} \in \Omega_h, \\ W(t, \mathbf{x}) = \Phi_{BC}(t, \mathbf{x}), & (t, \mathbf{x}) \in [0, +\infty[\times \partial\Omega_h. \end{cases} \tag{7.2}$$

with

$$\partial_t W + \mathrm{div} F^C(W) - \mathrm{div} F^D(W) = \partial_t W + \sum_{\ell=1}^{3} \partial_{x_\ell} F_\ell^C(W) - \sum_{\ell=1}^{3} \partial_{x_\ell} F_\ell^D(W) = F^S, \tag{7.3}$$

where, with the following notations $\mathbf{u} = (u_1, u_2, u_3)$, $\mathbf{u}' = (u_1', u_2', u_3')$,

$$W = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{pmatrix}, \quad F_\ell^C(W) = \begin{pmatrix} \rho u_\ell' \\ \rho u_1 u_\ell' + p\,\delta_{1\ell} \\ \rho u_2 u_\ell' + p\,\delta_{2\ell} \\ \rho u_3 u_\ell' + p\,\delta_{3\ell} \\ (\rho E + p) u_\ell' + p u_{e\ell} \end{pmatrix}, \quad F_\ell^D(W) = \begin{pmatrix} 0 \\ \mathcal{T}_{1\ell} \\ \mathcal{T}_{2\ell} \\ \mathcal{T}_{3\ell} \\ \sum_{i=1}^{3} u_i \mathcal{T}_{i\ell} + \lambda\, \partial_{x_\ell} T \end{pmatrix},$$

with

$$\mathcal{T}_{k\ell} = \mu \left(\partial_{x_\ell} u_k + \partial_{x_k} u_\ell - \frac{2}{3} \sum_{s=1}^{3} \partial_{x_s} u_s \delta_{k\ell} \right),$$

and the source terms fluxes is

$$F^S(W) = \begin{pmatrix} 0 \\ -\rho(\boldsymbol{\omega} \times \mathbf{u}) \\ 0 \end{pmatrix}.$$

Exactly as described in the Numerical Method Chapter, we have the following semi-discrete scheme,

$$\frac{dW_i}{dt} = \frac{1}{|C_i|} \left(\sum_{j \in \mathcal{V}(i)} \Phi_{ij}^C(W_i, W_j, \boldsymbol{n}_{ij}) + \sum_{K_i \in T(i)} \Phi_i^D|_{K_i}(W_i, W_j, W_k, W_l) + \Phi_i^S(W_i) \right).$$

where W_i stand for the following quantity

$$W_i = \frac{1}{|C_i|} \int_{C_i} W \, \mathbf{dx}.$$

As described before we have, with φ_i the P_1 Finite Element basis function associated with vertex P_i,

$$\Phi_i^D|_{K_i}(W_i, W_j, W_k, W_l) = -\int_{K_i} F^D(W)_{|K_i} \cdot \nabla \varphi_i \, \mathrm{d}x,$$

and

$$\Phi_i^S(W_i) = |C_i| \begin{pmatrix} 0 \\ -\omega \times (\rho \mathbf{u})_i \\ 0 \end{pmatrix}.$$

For the convective fluxes we consider the Roe approximate Riemann solver, which is described in the following subsection

$$\Phi^C(W_i, W_j, \mathbf{n}_{ij}) = \Phi^{\mathrm{Roe}}(W_i, W_j, \mathbf{n}_{ij}).$$

3.3 Roe scheme

Let us describe the Roe scheme for the convective part of the Navier-Stokes system in MRF formulation, while the diffusive part remains unchanged. Let us rewrite the convective part as follows

$$\partial_t W + \operatorname{div} F(W) = \partial_t W + \sum_{\ell=1}^{3} \partial_{x_\ell} F_\ell(W) = 0,$$

with

$$W = \begin{pmatrix} \rho \\ \rho \mathrm{u}_1 \\ \rho \mathrm{u}_2 \\ \rho \mathrm{u}_3 \\ \rho E \end{pmatrix}, \quad F_\ell(W) = \begin{pmatrix} \rho(\mathrm{u}_\ell - \mathrm{u}_{e,\ell}) \\ \rho \mathrm{u}_1(\mathrm{u}_\ell - \mathrm{u}_{e,\ell}) + \delta_{\ell 1} p \\ \rho \mathrm{u}_2(\mathrm{u}_\ell - \mathrm{u}_{e,\ell}) + \delta_{\ell 2} p \\ \rho \mathrm{u}_3(\mathrm{u}_\ell - \mathrm{u}_{e,\ell}) + \delta_{\ell 3} p \\ \rho E(\mathrm{u}_\ell - \mathrm{u}_{e,\ell}) + p \mathrm{u}_\ell \end{pmatrix}.$$

Note that the flux can be broken down into an Euler flux and another flux which we will call the MRF flux,

$$F_\ell(W) = F_\ell^{\text{Euler}}(W) + F_\ell^{\text{MRF}}(W),$$

where

$$F_\ell^{\text{MRF}}(W) = -\rho \mathrm{u}_{e,\ell} \begin{pmatrix} 1 \\ \mathrm{u}_1 \\ \mathrm{u}_2 \\ \mathrm{u}_3 \\ E \end{pmatrix} = -\mathrm{u}_{e,\ell}\, W.$$

So the Jacobian matrix of our flux is

$$\begin{aligned} \nabla_W F_\ell(W) &= \nabla_W F_\ell^{\text{Euler}}(W) + \nabla_W F_\ell^{\text{MRF}}(W) \\ &= A_\ell^{\text{Euler}}(W) + A_\ell^{\text{MRF}}(W), \end{aligned}$$

where A_ℓ^{Euler} are the Jacobian matrices of the Euler flux and

$$A_\ell^{\text{MRF}}(W) = -\mathrm{u}_{e,\ell}\, \mathbb{I}_5.$$

As $A_\ell^{\text{MRF}}(W)$ are diagonal matrices it is easy to see that the Jacobian matrix of our flux is digonalizable and we can quickly find its eigenvalues, indeed

$$\begin{aligned}
A_\ell(W) = \nabla_W F_\ell(W) &= A_\ell^{\text{Euler}}(W) + A_\ell^{\text{MRF}}(W) \\
&= T \begin{pmatrix} \lambda_\ell^{(1)} & 0 & 0 & 0 & 0 \\ 0 & \lambda_\ell^{(2)} & 0 & 0 & 0 \\ 0 & 0 & \lambda_\ell^{(3)} & 0 & 0 \\ 0 & 0 & 0 & \lambda_\ell^{(4)} & 0 \\ 0 & 0 & 0 & 0 & \lambda_\ell^{(5)} \end{pmatrix} T^{-1} - \mathrm{u}_{e,\ell}\, \mathbb{I}_5 \\
&= T \begin{pmatrix} \lambda_\ell^{(1)} - \mathrm{u}_{e,\ell} & 0 & 0 & 0 & 0 \\ 0 & \lambda_\ell^{(2)} - \mathrm{u}_{e,\ell} & 0 & 0 & 0 \\ 0 & 0 & \lambda_\ell^{(3)} - \mathrm{u}_{e,\ell} & 0 & 0 \\ 0 & 0 & 0 & \lambda_\ell^{(4)} - \mathrm{u}_{e,\ell} & 0 \\ 0 & 0 & 0 & 0 & \lambda_\ell^{(5)} - \mathrm{u}_{e,\ell} \end{pmatrix} T^{-1},
\end{aligned}$$

the eigenvalues of $A_\ell(W)$ are consequently

$$\lambda_\ell^{(1)}(W) = \mathrm{u}_{r,\ell} - c,\ \ \lambda_\ell^{(2)}(W) = \lambda_\ell^{(3)}(W) = \lambda_\ell^{(4)}(W) = \mathrm{u}_{r,\ell},\ \ \lambda_\ell^{(5)}(W) = \mathrm{u}_{r,\ell} + c.$$

The jump conditions are also trivial to check, we have $\left(\text{with } \overline{W}_{ij} = (\rho_i^{\frac{1}{2}} W_i + \rho_j^{\frac{1}{2}} W_j)/(\rho_i^{\frac{1}{2}} + \rho_j^{\frac{1}{2}})\right)$

$$\begin{gathered}
F_\ell(W_i) - F_\ell(W_j) = A_\ell(\overline{W}_{ij})(W_i - W_j) \\
\Longleftrightarrow \\
\underbrace{F_\ell^{\text{Euler}}(W_i) - F_\ell^{\text{Euler}}(W_j) - A_\ell^{\text{Euler}}(\overline{W}_{ij})(W_i - W_j)}_{=0} = \underbrace{F_\ell^{\text{MRF}}(W_i) - F_\ell^{\text{MRF}}(W_j) - A_\ell^{\text{MRF}}(\overline{W}_{ij})(W_i - W_j)}_{=0}.
\end{gathered}$$

Finally, our Roe fluxes are given by

$$\begin{aligned}
\Phi_\ell^{\text{Roe}}(W_i, W_j, \mathbf{n}_{ij}) &= \frac{1}{2}\left(F_\ell(W_i) + F_\ell(W_j)\right) \cdot \mathbf{n}_{ij} + \frac{1}{2}|A_\ell(\overline{W}_{ij})|\,(W_i - W_j) \\
&= \frac{1}{2}\left(F_\ell^{\text{Euler}}(W_i) - (\mathbf{u}_{e,\ell})_{ij} W_i + F_\ell^{\text{Euler}}(W_j) - (\mathbf{u}_{e,\ell})_{ij} W_j\right) \cdot \mathbf{n}_{ij} \\
&\quad + \frac{1}{2}|A_\ell^{\text{Euler}}(\overline{W}_{ij}) - (\mathbf{u}_{e,\ell})_{ij}|\,(W_i - W_j)\,.
\end{aligned}$$

In fact, our MRF formulation comes down to an ALE formulation without mesh deformation ! It is interesting to use an ALE method that has already been implemented for the MRF method.

3.4 Discrete Geometric Conservation Law for MRF case

In the NiceFlow code, an ALE method is already implemented, so we use this method to implement our MRF method. Let us briefly review this method and show how it applies to our case.

3.4.1 Notations for MRF case

We consider only the convective part of our equation in MRF formulation, which read

$$\partial_t W + \operatorname{div} F^C(W) = 0, \tag{7.4}$$

with

$$F^C(W) = F(W) + F^{MRF}(W) = \begin{pmatrix} \rho\mathbf{u} \\ \rho\,\mathbf{u}\otimes\mathbf{u} + p\mathbb{I}_3 \\ (\rho E + p)\mathbf{u} \end{pmatrix} - \begin{pmatrix} \rho\mathbf{u}_e \\ \rho\,\mathbf{u}_e\otimes\mathbf{u} \\ \rho E\mathbf{u}_e \end{pmatrix}$$

Now integrating (7.4) on each cells C_i we have

$$|C_i|\frac{dW_i}{dt} + \sum_{j\in V(i)} \int_{\partial C_{ij}} \left(F(W)\cdot\mathbf{n}_i - W\,\mathbf{u}_e\cdot\mathbf{n}_i\right)\,\mathrm{d}s = 0,$$

and we consider the following approximation

$$\int_{\partial C_{ij}} \left(F(W)\cdot\mathbf{n}_i - W\,\mathbf{u}_e\cdot\mathbf{n}_i\right)\,\mathrm{d}s \approx |\partial C_{ij}|\Phi(W_i, W_j, \mathbf{n}_{ij}, \sigma_{ij})$$

where

$$\begin{aligned} \mathbf{n}_{ij} &= \frac{1}{|\partial C_{ij}|}\int_{\partial C_{ij}} \mathbf{n}_i\,\mathrm{d}s, \\ \sigma_{ij} &= \frac{1}{|\partial C_{ij}|}\int_{\partial C_{ij}} \mathbf{u}_e\cdot\mathbf{n}_i\,\mathrm{d}s, \end{aligned}$$

and moreover

$$\begin{aligned} &\Phi(W_*, W^*, \mathbf{n}, \sigma) = -\Phi(W^*, W_*, -\mathbf{n}, \sigma), \quad &\text{(conservativity)} \\ &\Phi(W, W, \mathbf{n}, \sigma) = F(W)\cdot\mathbf{n} - W\sigma. &\text{(consistency)} \end{aligned}$$

3.4.2 Discretization of moving domains : ALE method

Let us consider a smooth bijective mapping $\pi(t)$ depending on time and equal to identity at time $t = 0$. Defining this mapping is equivalent to defining a velocity field $\mathbf{u}_e$ and moving each point of the space with this velocity. Let for any time $\Omega_{\mathrm{h}}(t)$ be the triangulation (tetrahedrisation in 3D) obtained by applying the mapping $\pi(t)$ to any vertex of Ω_{h}^0. It is enough to known the trajectories of each vertex, starting from a vertex of Ω_{h}^0. This is also equivalent to know for any time t the velocity $\mathbf{u}_e(i,t)$ at vertex i of Ω_{h}^t. In order to ensure than any segment or plan of the initial mesh will stay at each time resp. a segment or a plan, we consider the linear interpolation of vertex values $\mathbf{u}_e(i,t)$ to any element of mesh Ω_{h}^t. For this discrete deformation velocity, we keep the notation $\mathbf{u}_e(t)$. In Ω_{h}^t we build the dual cells :

$$\Omega_{\mathrm{h}}(t) = \bigcup_{i=1}^{n_c} C^i(t).$$

The flux balance writes now :

$$\frac{d\mathcal{V}^i}{dt} \ + \ \int_{\partial C^i(t)} \boldsymbol{F}(v)\cdot\mathbf{n}_i \, \mathrm{d}s \ - \ \int_{\partial C^i(t)} v\, \mathbf{u}_e \cdot \mathbf{n}_i \, \mathrm{d}s = 0, \tag{7.5}$$

where

$$\frac{d\mathcal{V}^i}{dt} = \frac{d}{dt}\left(\int_{C^i(t)} v\right).$$

For each *interior* cell $C^i(t)$, with its corresponding set $V(i)$ of neighboring cells,

$$\partial C^i(t) = \bigcup_{j\in V(i)} \partial C^{ij}(t), \tag{7.6}$$

where $\partial C^{ij}(t)$ represents the interface shared by cells $C^i(t)$ and $C^j(t)$. Then, (7.5) can be written as :

$$\frac{d\mathcal{V}^i}{dt} \ + \sum_{j\in V(i)} \int_{\partial C^{ij}(t)} \boldsymbol{F}(v)\cdot\mathbf{n}_{ij} \, \mathrm{d}s \ - \sum_{j\in V(i)} \int_{\partial C^{ij}(t)} v\, \mathbf{u}_e \cdot \mathbf{n}_{ij} = 0 \, \mathrm{d}s. \tag{7.7}$$

In order to evaluate this integral, following [45], let us define

$$\boldsymbol{\nu}_{ij}(t) \ = \ \frac{1}{|\partial C_{ij}(t)|} \int_{\partial C_{ij}(t)} \mathbf{n}_{ij}(t) \, \mathrm{d}s \tag{7.8}$$

and

$$\kappa_{ij}(t) \ = \ \frac{1}{|\partial C_{ij}(t)|} \int_{\partial C_{ij}(t)} \mathbf{u}_e(t) \ \cdot \ \mathbf{n}_{ij}(t) \, \mathrm{d}s \ . \tag{7.9}$$

$\boldsymbol{\nu}_{ij}(t)$ is the *mean normal vector* corresponding to cell interface $\partial C_{ij}(t)$ and $\kappa_{ij}(t)$, the *mean normal mesh velocity* projection for the same cell interface (the full meaning of this “mean” will be grasped below, when it becomes also a temporal one). Then we get an integral ALE semi-discretization of the conservation law :

$$\frac{d\mathcal{V}^i}{dt} \ + \sum_{j\in V(i)} |\partial C_{ij}(t)| \ \Phi(v_i, v_j, \boldsymbol{\nu}_{ij}(t), \kappa_{ij}(t)) \ = \ 0, \tag{7.10}$$

where Φ is a numerical flux function, typically an approximate Riemann solver, with mesh velocity normal component $\kappa_{ij}(t)$ and with mean value of unknown v over cell i denoted by v_i. In particular it satisfies the following consistency condition :

$$\Phi(v, v, \boldsymbol{\nu}, \kappa) \;=\; \boldsymbol{F}(v) \;\cdot\; \boldsymbol{\nu} \;-\; \kappa v.$$

Consider now a time discretization of the above formula. Up to first order, v_i can be taken as constant within each cell. Then, if the volume of the partition's cell $C_i(t)$ is $|C_i(t)|$,

$$\mathcal{V}_i(t) = |C_i(t)|\, v_i(t). \tag{7.11}$$

The θ-parameterized Euler time advancing yields

$$\begin{aligned} |C_i^{n+1}| v_i^{n+1} \;&=\; |C_i^n| v_i^n \\ &\quad - \Delta t \theta \sum_{j \in V(i)} |\partial \bar{C}_{ij}| \Phi \left(v_i^{n+1}, v_j^{n+1}, \overline{\boldsymbol{\nu}}_{ij}, \overline{\kappa}_{ij} \right) \\ &\quad - \Delta t (1-\theta) \sum_{j \in V(i)} |\partial \bar{C}_{ij}| \Phi \left(v_i^n, v_j^n, \overline{\boldsymbol{\nu}}_{ij}, \overline{\kappa}_{ij} \right) \end{aligned} \tag{7.12}$$

where the overlines mean that time averaged values are taken. According to the Geometric Conservation Law principle, a uniform solution is exactly preserved when time-advanced by the numerical scheme. Assume the above system able to reproduce a constant solution $v^n = v^{n+1} = v^*$, it should satisfy :

$$|C_i^{n+1}| v_i^* \;=\; |C_i^n| v_i^* \;-$$

$$\Delta t \theta \sum_{j \in V(i)} |\partial \bar{C}_{ij}| \Phi \left(v_i^*, v_j^*, \overline{\boldsymbol{\nu}}_{ij}, \overline{\kappa}_{ij} \right) \;-\; \Delta t (1-\theta) \sum_{j \in V(i)} |\partial \bar{C}_{ij}| \Phi \left(v_i^*, v_j^*, \overline{\boldsymbol{\nu}}_{ij}, \overline{\kappa}_{ij} \right).$$

Invoking the consistency condition for Φ and the fact that the cells remain closed during the motion, which writes :

$$\sum_{j \in V(i)} |\partial \bar{C}_{ij}| \overline{\boldsymbol{\nu}}_{ij} \;=\; 0,$$

we see that this gives the usual *Discrete Geometric Conservation Law (DGCL)* :

$$|C_i^{n+1}| \;=\; |C_i^n| \;-\; \Delta t \sum_{j \in V(i)} |\partial \bar{C}_{ij}| \overline{\kappa}_{ij}. \tag{7.13}$$

As stated in papers like [99], [44] or [40], the DGCL becomes a design condition to impose for the time averaged values $|\partial \bar{C}_{ij}|$, $\overline{\boldsymbol{\nu}}_{ij}$, and $\overline{\kappa}_{ij}$. Evaluated over $\partial C^{ij}(t)$, they should be carefully computed. In [99], both the cell's normals and the grid's velocity mean values determine the geometrical parameters which enforce the GCL. In [44], this is attained by means of a proper evaluation of the ALE fluxes using suited mesh configurations and grids velocities, showing also an equivalence with the former paper ideas. On the other hand, in [40], a scheme satisfying the GCL is proposed by tuning how the cell volume is evaluated. We also refer to [45] for examples of averagings satisfying the DGCL for the above time advancing scheme.

3.4.3 Adaptation of DGCL to MRF

Following the previous lines, but for a domain that does not change shape over time, the last result gives us the Discrete Geometric Conservation Law for the MRF formulation

$$\sum_{j\in V(i)} |\partial C_{ij}|\, \sigma_{ij} = 0,$$

which can be rewritten

$$\sum_{j\in V(i)} |\partial C_{ij}|\, \sigma_{ij} = \sum_{j\in V(i)} \int_{\partial C_{ij}} \mathbf{u}_e \cdot \mathbf{n}_{ij}\, \mathrm{d}s = \int_{\partial C_i} \mathbf{u}_e \cdot \mathbf{n}_i\, \mathrm{d}s = \int_{C_i} \mathrm{div}(\mathbf{u}_e)\, \mathrm{d}\mathbf{x} = 0.$$

As concerns the DGCL inside MRF, we observe that the mesh is fictitiously rotating in one part Ω_r of the computational domain and fixed in the rest of the domain. Then the fictitous mesh speed $\mathbf{u}_e$ is discontinuous, typically :

$$\mathbf{u}_e = \boldsymbol{\omega} \times \mathbf{x} \quad \text{in} \quad \Omega_r, \quad 0 \quad \text{elsewhere.}$$

However, the velocity is of zero divergence. Indeed the divergence is zero on both subdomains, and the normal velocity to interface is zero, which implies that the divergence is zero over the complete computational domain. In the discrete case, the DGCL is easily checked in both domain. In the neighborhood of the discontinuity, we have checked that in practice, the numerical deviation to DGCL is close to zero machine.

4 Boundary conditions

For both the MRF and ALE methods, the boundary conditions are the same as for the formulation of the standard Navier-Stokes equations, except for slipping boundary conditions :

$$\Phi_{Slip} = F^C(W) \cdot \mathbf{n} = \begin{pmatrix} 0 \\ p\,\mathbf{n} \\ -p\,\mathbf{u}_e \cdot \mathbf{n} \end{pmatrix}.$$

5 First numerical application : Caradonna-Tung (1981) model helicoper rotor in hover

As a first experiment to validate our method with mesh adaptation, we chose the Caradonna-Tung test case. A positive aspect of this test case is the large amount of experimental data available. This experiment was initiated by F. X. Caradonna and C.

Tung in 1981 in [29] and consists of a rotor made up of two rectangular blades of profile NACA0012.

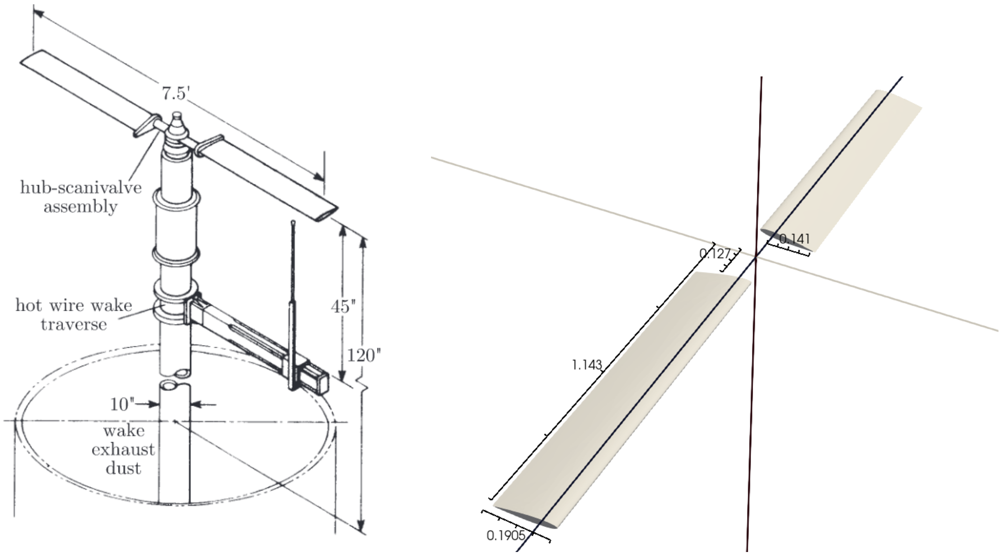


FIGURE 7.9 – Experimental set-up of Caradonna-Tung two-bladed model rotor in hover [29] (left).

For rotor dimensions, the chord length of a blade is $0.1905\,m$ and the rotor diameter is $1.143\,m$. Moreover the blade angle is fixed at $8°$ and the blade is eccentric with respect to the axis of rotation, so that the distance between the back of the blade and our reference frame, calculated along the chord, is $0.141\,m$.

For the simulations, the axial flow mode was selected at a rotor speed of $650\,RPM$, which corresponds to the blade tip velocity of $77.8\,m/s$ and the tip Mach number = 0.228.

5.1 Steady case

The computation was carried out using the Spalart-Almaras RANS turbulence model, with the Roe scheme, V4 reconstruction and a parameter $\gamma = 0.3$ to control the numerical viscosity. The time part is solved in local time stepping using the implicit BDF1 scheme, with a CFL fixed at 2. Given that we are interested in a RANS flow and that our method for taking rotation into account is a stationary method, we will use the stationary mesh adaptation method for this case.

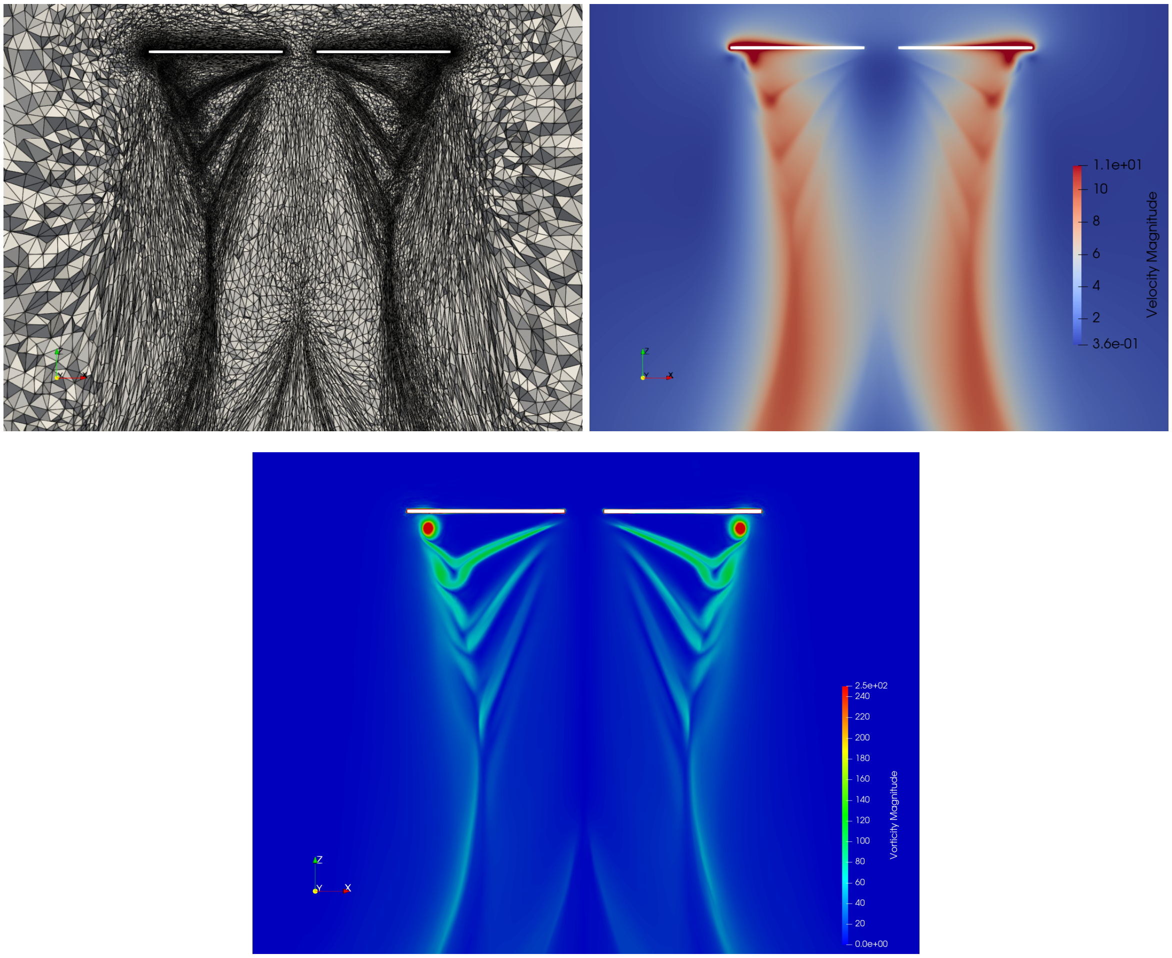


FIGURE 7.10 – Caradonna-Tung RANS mesh adaptive simulation results : mesh (top left), velocity field (top right) and vorticity field (bottom) in cross-section.

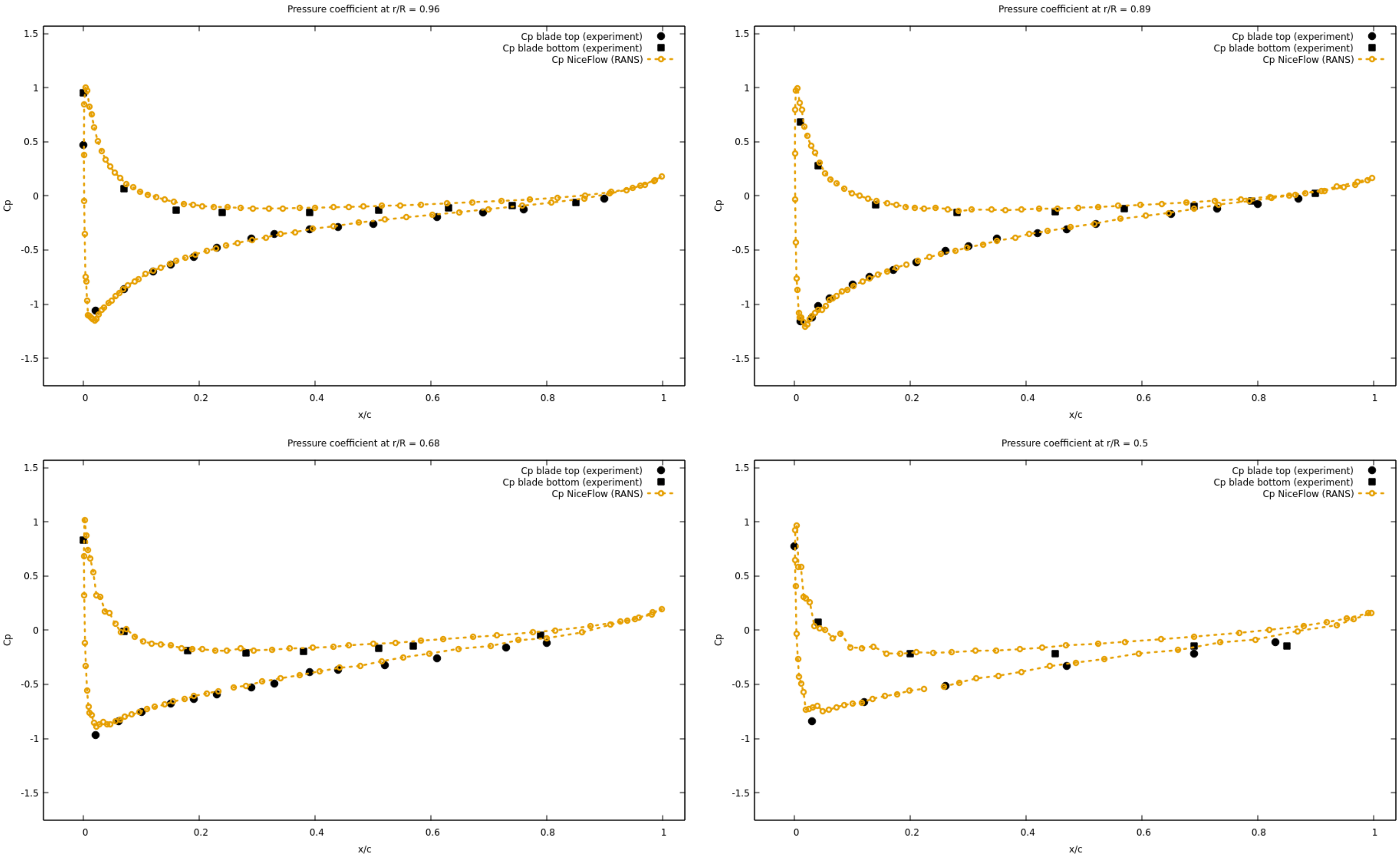


FIGURE 7.11 – **Top** : pressure coefficient at $r/R = 0.96$ (left) and $r/R = 0.89$ (right) blade section. **Bottom** : pressure coefficient at $r/R = 0.68$ (left) and $r/R = 0.5$ (right) blade section.

The results presented were obtained after 33 remeshing iterations, for a final adapted mesh of around 3.5 million vertices, see Figure 7.10. Figure 7.11 represent the pressure coefficient distribution along the blade at specific section locations, a good agreement with experimental data can be observed.

In Figure 7.13, the tip vortex position is compared against experimental data and analytical model prediction [72] The tip vortex contraction (r/R) predicted in RANS calculation is well aligned with predicted tip vortex position for vortex age, but not close to the experimental data. As for the wake descent (z/R), computational result is close to the experiment and predicted tip vortex position for vortex age up to $250°$.

Three comments on this result : firstly, this calculation is a RANS calculation, a DES type calculation would surely give better results. Secondly, we used the V4 reconstruction instead of V6 for practical reasons, whereas V6 would have given a better prediction for Tip vortex position. Finally, processing the vorticity fields to obtain the position of these vortexes can lead to inaccuracies.

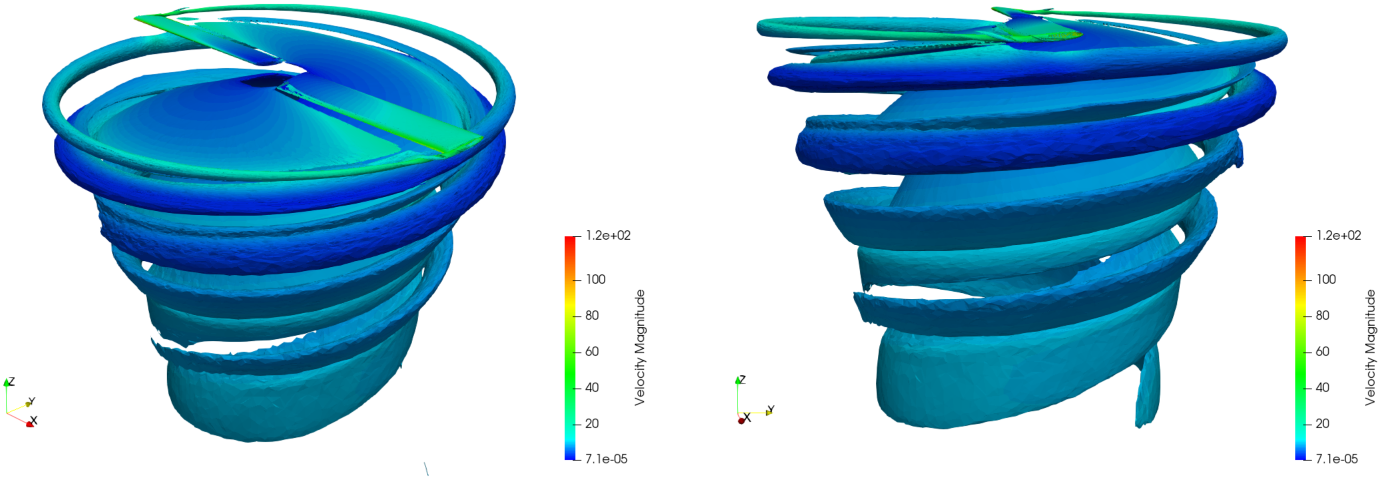


FIGURE 7.12 – Caradonna-Tung RANS mesh adaptive simulation results : Q-criterion iso-surface.

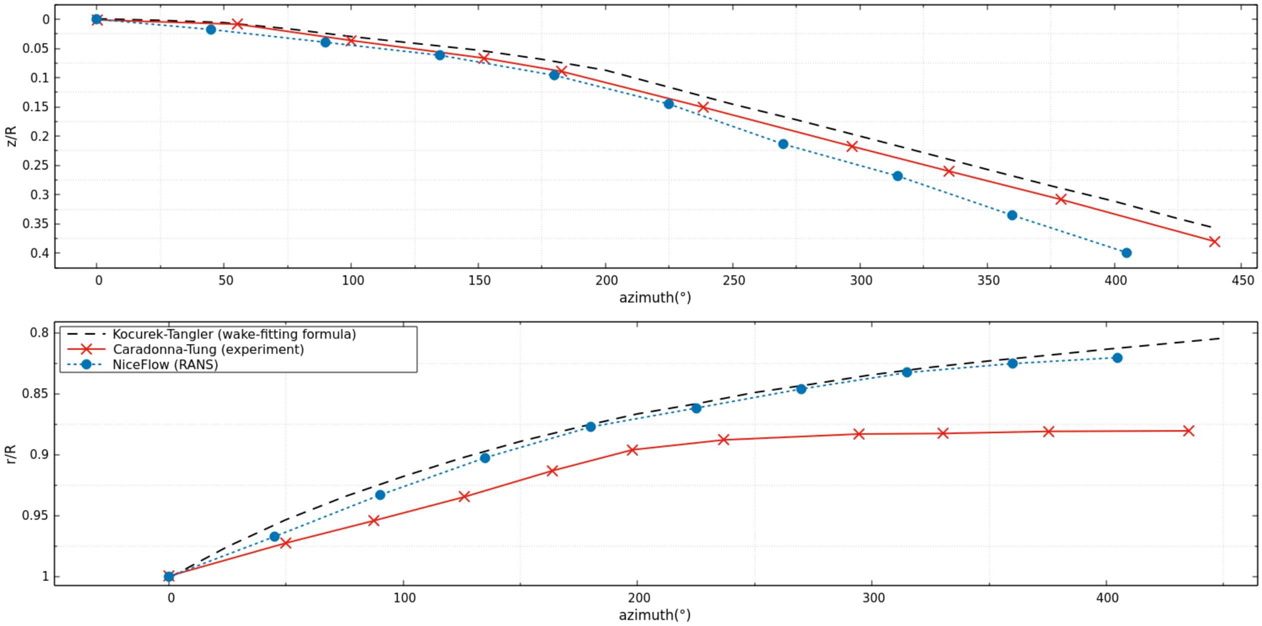


FIGURE 7.13 – Tip vortex position.

5.2 Unsteady case

The aim of this computation is to capture the unsteadiness produced by the turbulence generated by our rotor model. For this, we use the DDES turbulence model and the

Transient Fixed Point unsteady mesh adaptation algorithm as follows :
Mesh adaptation is performed every $0.23s$ at the start of the computation, when the chosen complexity is not yet too great, and every $0.00575s$ thereafter, with one fixed point each time.

For the sake of clarity, let's remember that our rotor speed is $650rpm$, so a time of $0.23s$ corresponds to about $1/4$ of rotor revolution and 0.00575 to about $1/16$. And the final computation time is $0.2875s$ which corresponds to about 3 turns and $1/8$ of the rotor.

For the numerical conditions of the simulation, we are in global time stepping, implicit, with the BDF2 scheme. The finite volume space part is handled by the Roe solver, with a parameter $\gamma = 0.3$ to control the numerical viscosity. The V4 reconstruction is still used instead of V6 for reasons of stability and computational progress. Indeed, the size of the smallest elements is of the order of 10^{-7}, which doesn't make the calculation any easier in terms of time, and poses stability problems. The CFL chosen to advance the computation is 250. It should be pointed out that this CFL value was lowered during the calculation and mesh adaptations.

The final mesh shown contains around $20.3\ M$ vertices. By comparing the results of the Figures 7.14 - 7.16 and 7.10, 7.12, a clear improvement can be seen, particularly in the capture of vortex wakes. It should also be noted that the objective of this computation has been achieved (partially), as can be seen in Figure 7.17, we have succeeded in capturing small structures generated by our blades. The pressure coefficient on the blade surface is as good as before (Figures 7.18-7.21), with a slight improvement close to the center of rotation (Figures 7.20 and 7.21).

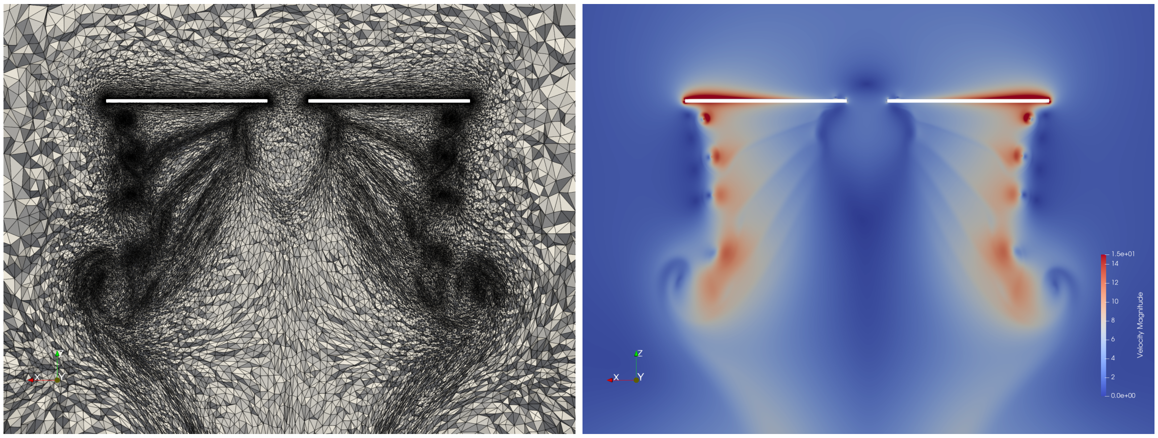


FIGURE 7.14 – Caradonna-Tung DDES simulation results : mesh (left) and velocity field (right) in cross-section.

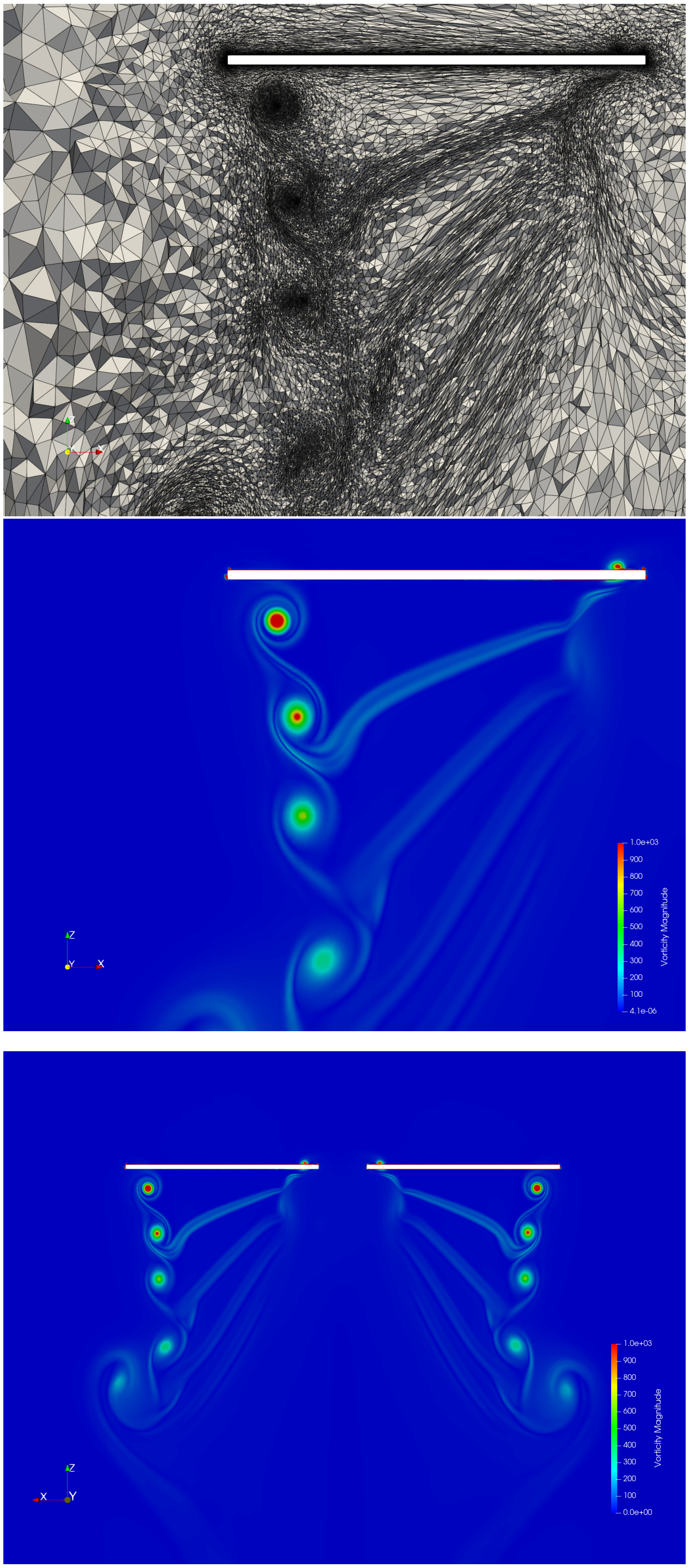
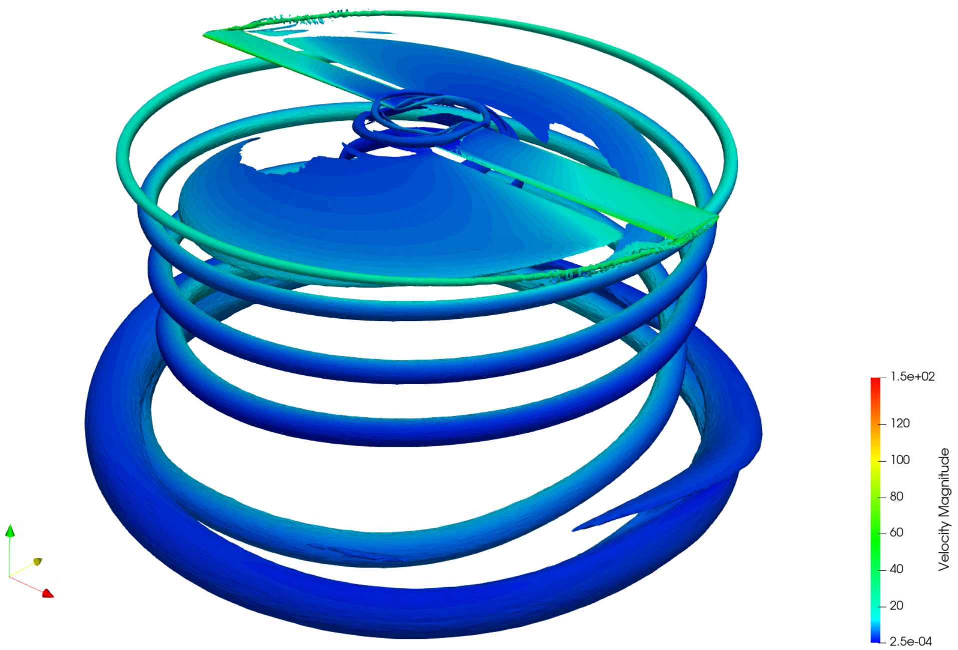


Figure 7.15 – Caradonna-Tung DDES simulation results : mesh (top) and vorticity field (bottom) in cross-section.



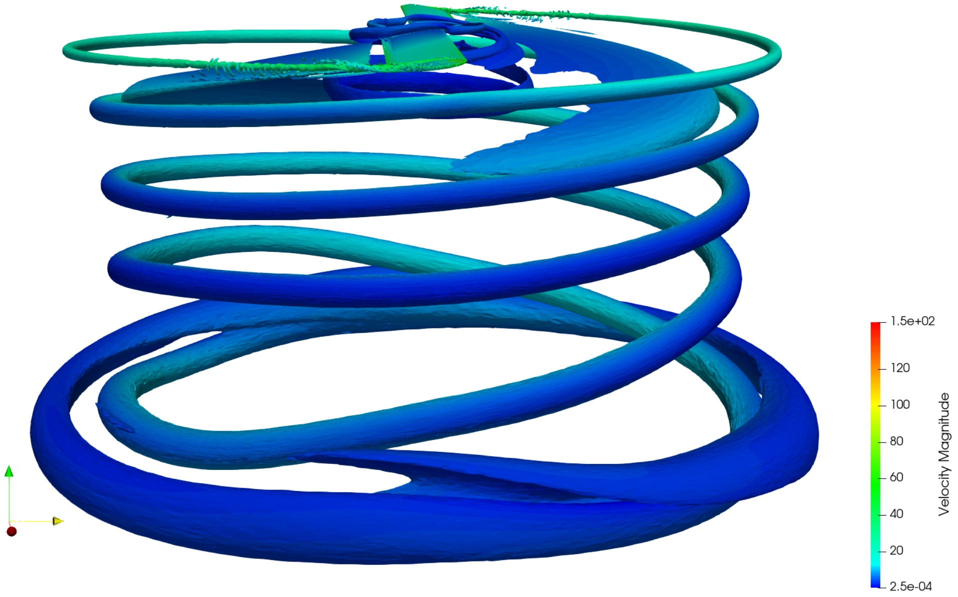


FIGURE 7.16 – Caradonna-Tung DDES simulation results : Q-criterion iso-surface.

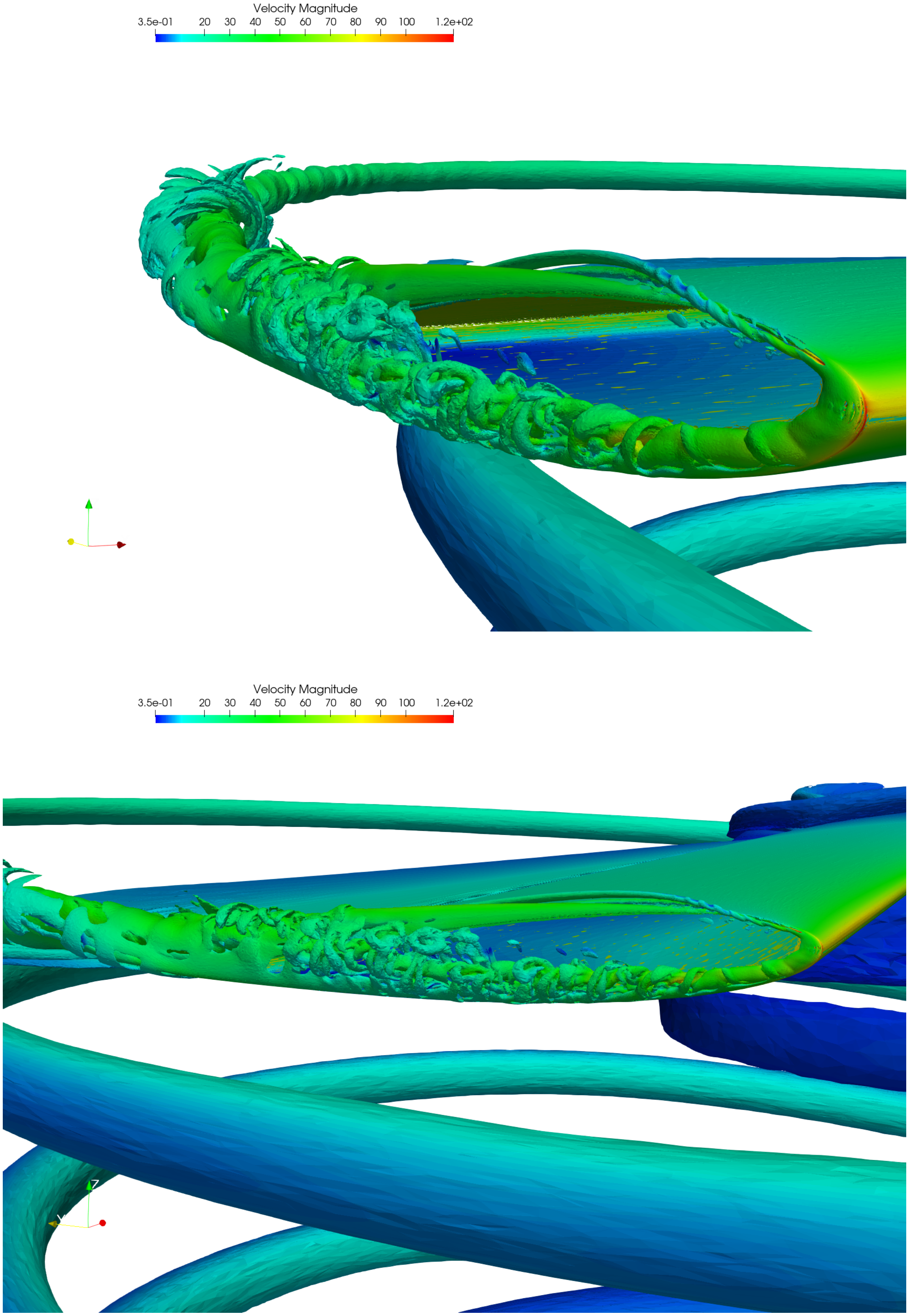


FIGURE 7.17 – Caradonna-Tung DDES simulation results : Q-criterion iso-surface (zoom).

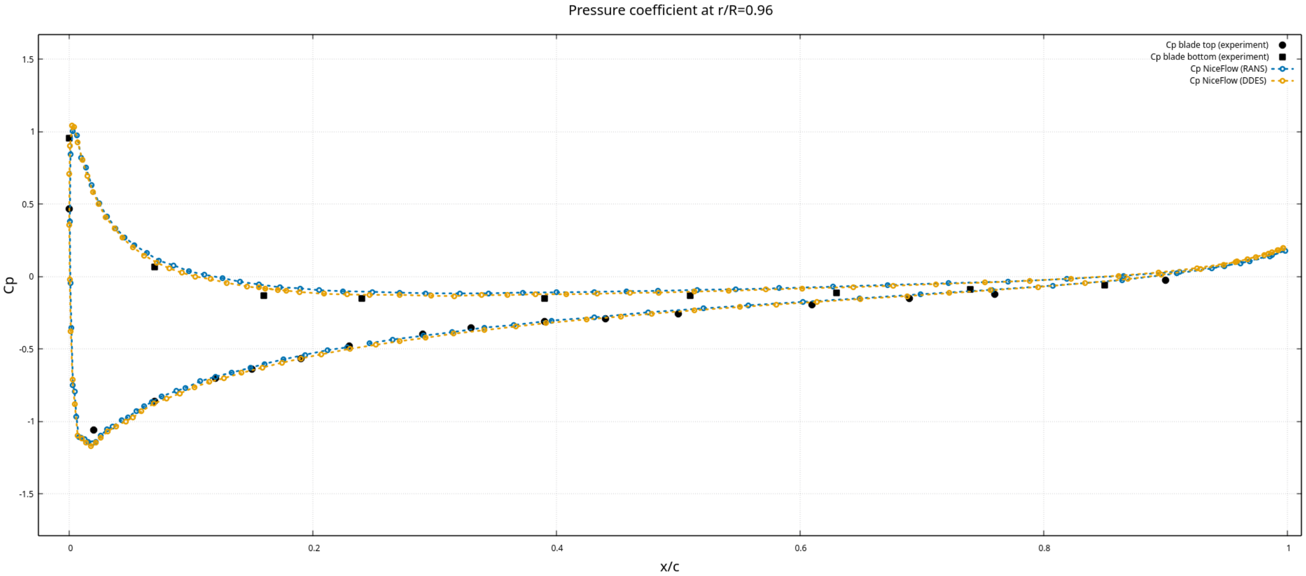


FIGURE 7.18 – Pressure coefficient at $r/R = 0.96$.

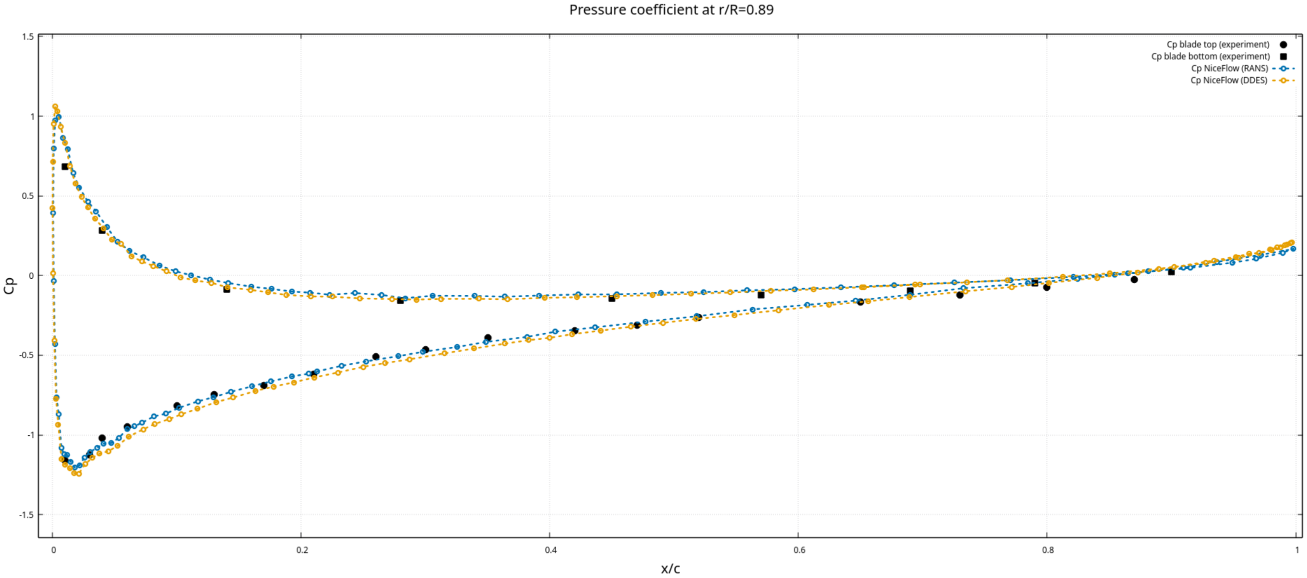


FIGURE 7.19 – Pressure coefficient at $r/R = 0.89$.

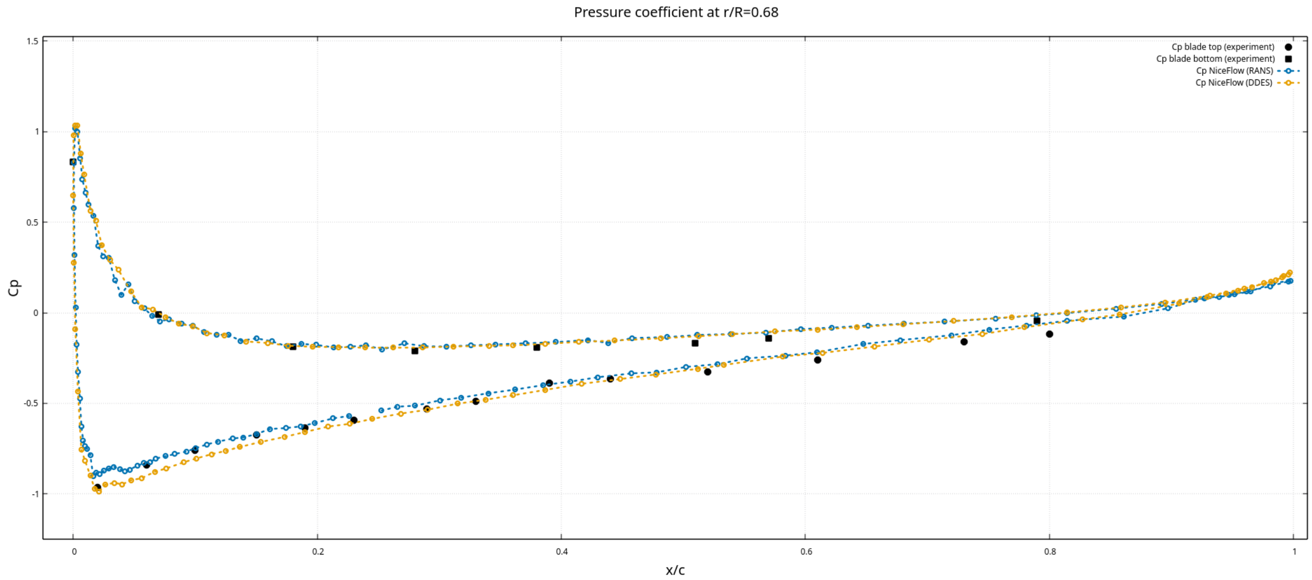


FIGURE 7.20 – Pressure coefficient at $r/R = 0.68$.

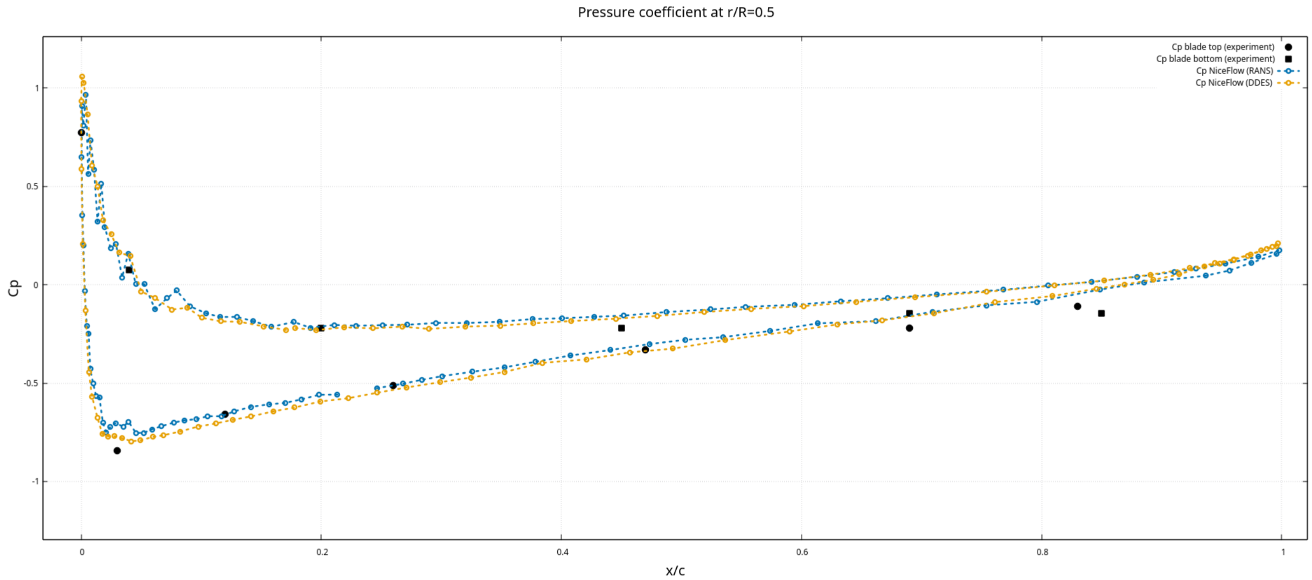


FIGURE 7.21 – Pressure coefficient at $r/R = 0.5$.

5.3 First Helicopter model in hover

We present our results for our first simulation test with rotor-fuselage interaction. This model was inspired by the experience of the following technical report [47] but adapted to our case, i.e. the rotor part is our Caradonna-Tung model studied previously, combined with a Robin fuselage. The fuselage dimensions can be found in [47] and are partially shown in Figure 7.22 .

This is a simulation of a simplified model of a hovering helicopter, the rotor speed is always $650rpm$. Unfortunately we have no experimental data for this test case, so the values given in Figure 7.25-7.30 are indicative. However, there are experimental values for a non-stationary flight case, with a forward speed of $40\,m/s$.

The simulation parameters are the same as for stationary Caradonna-Tung, i.e. the Spalart-Almaras turbulence model, the Roe scheme with $\gamma = 0.3$ and the second-order V4 reconstruction are used. The time part is solved in local time stepping using the implicit BDF1 scheme, with a CFL fixed at 1. We use the stationary mesh adaptation method.

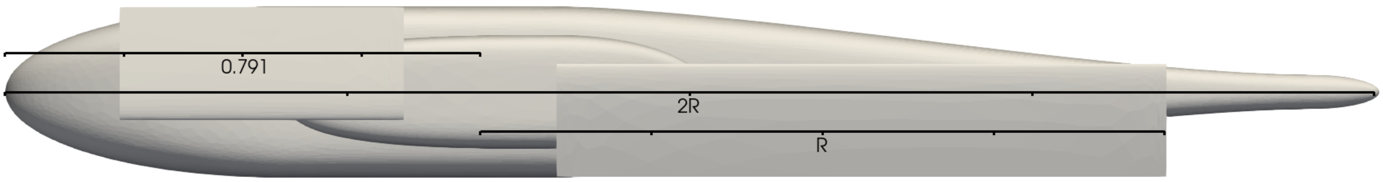


FIGURE 7.22 – Fuselage size compared to rotor length.

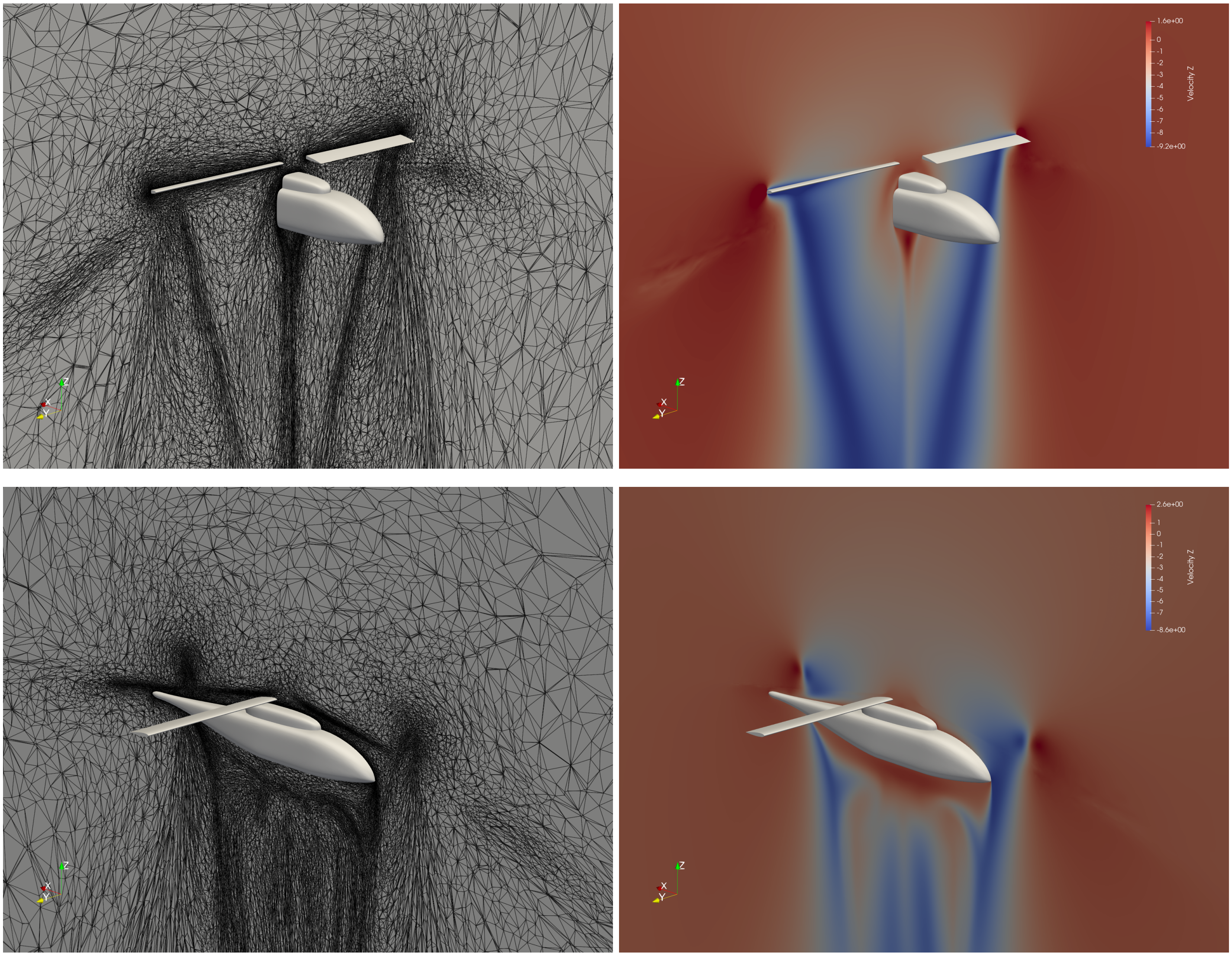


FIGURE 7.23 – Robin RANS simulation results : mesh (left) and velocity field (right) in cross-section.

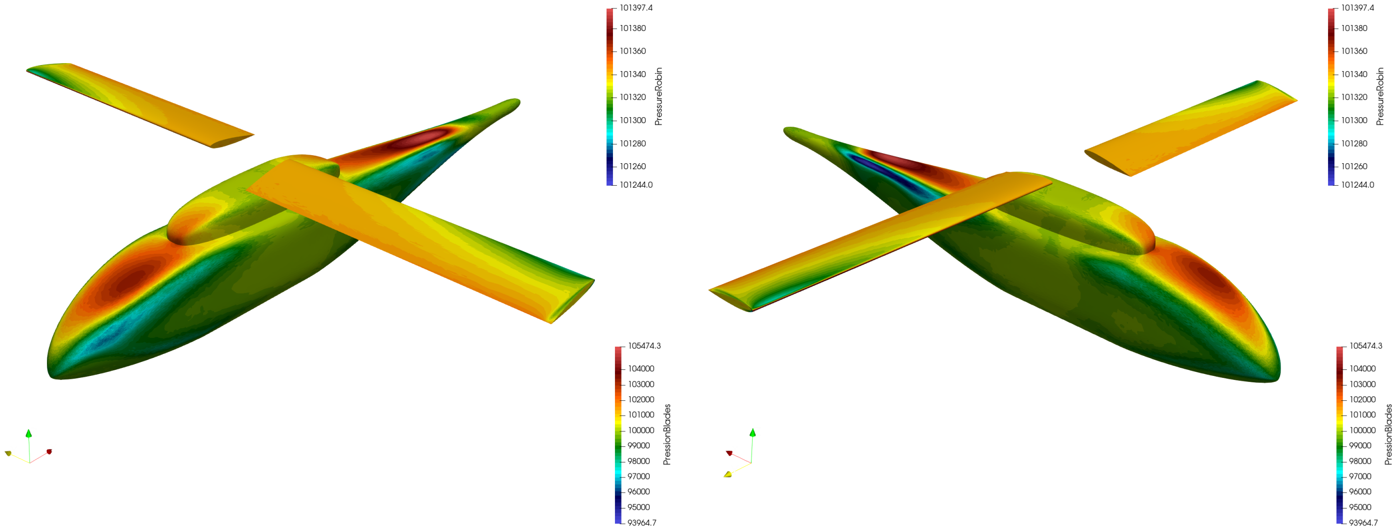


FIGURE 7.24 – Robin RANS simulation results : Surface pressure.

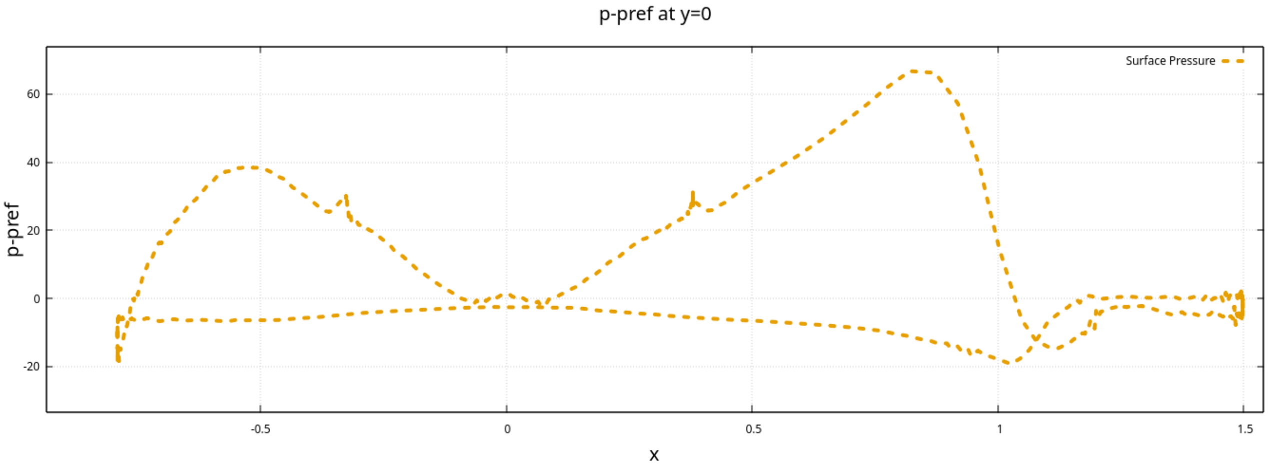


FIGURE 7.25 – Robin pressure at $y = 0$.

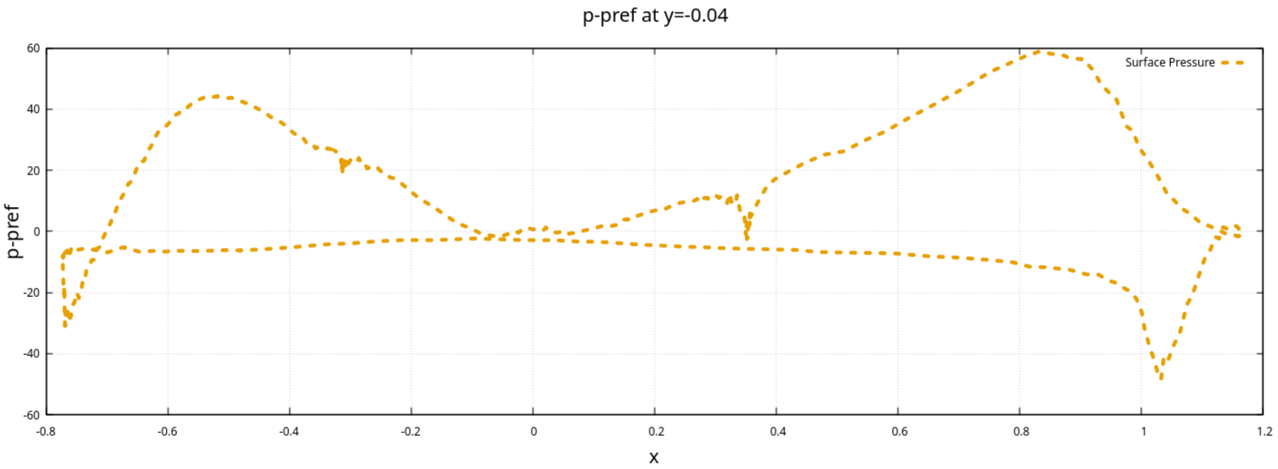


FIGURE 7.26 – Robin pressure at $y = -0.04$.

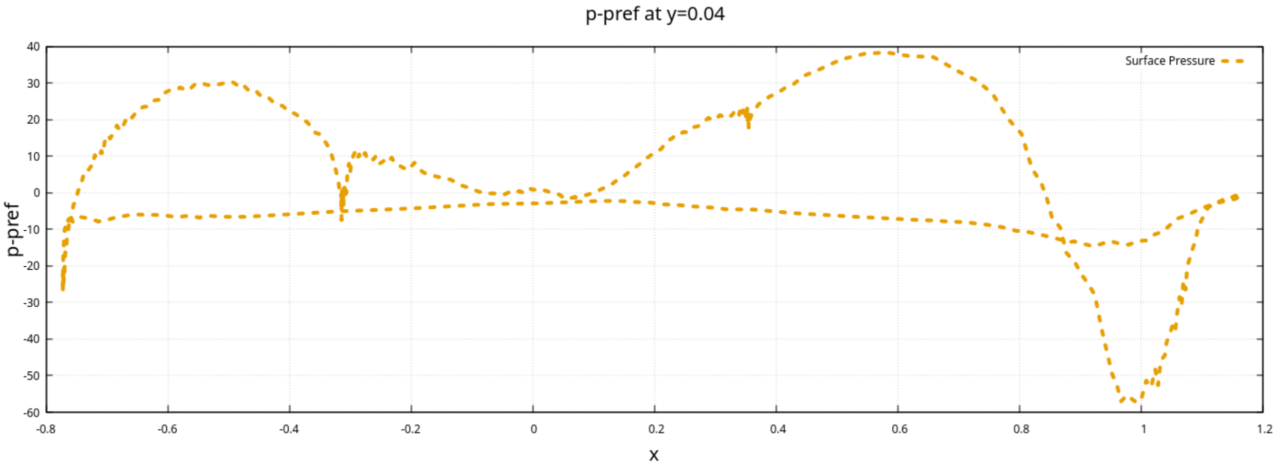


FIGURE 7.27 – Robin pressure at $y = 0.04$.

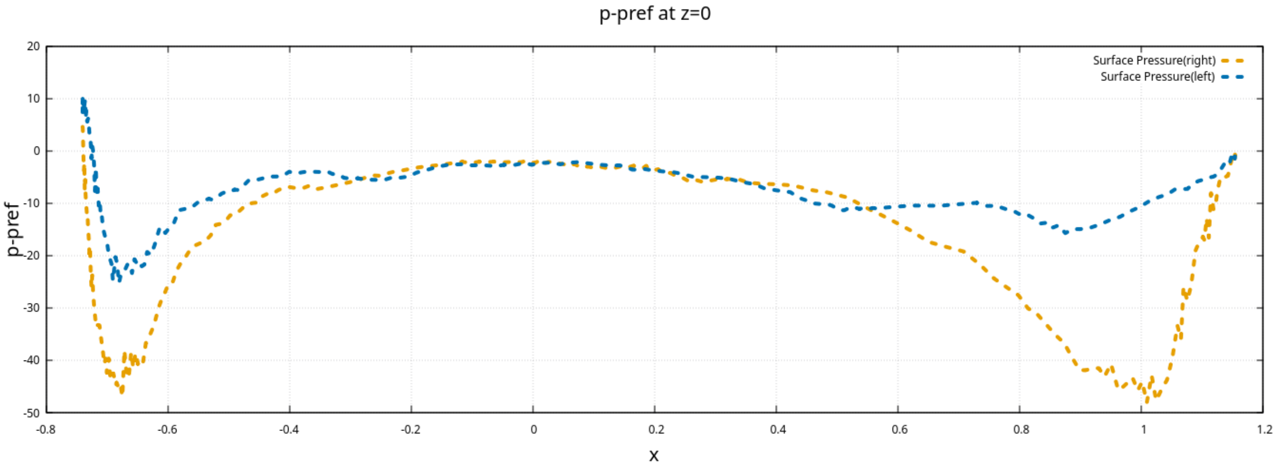


Figure 7.28 – Robin pressure at $z = 0$.

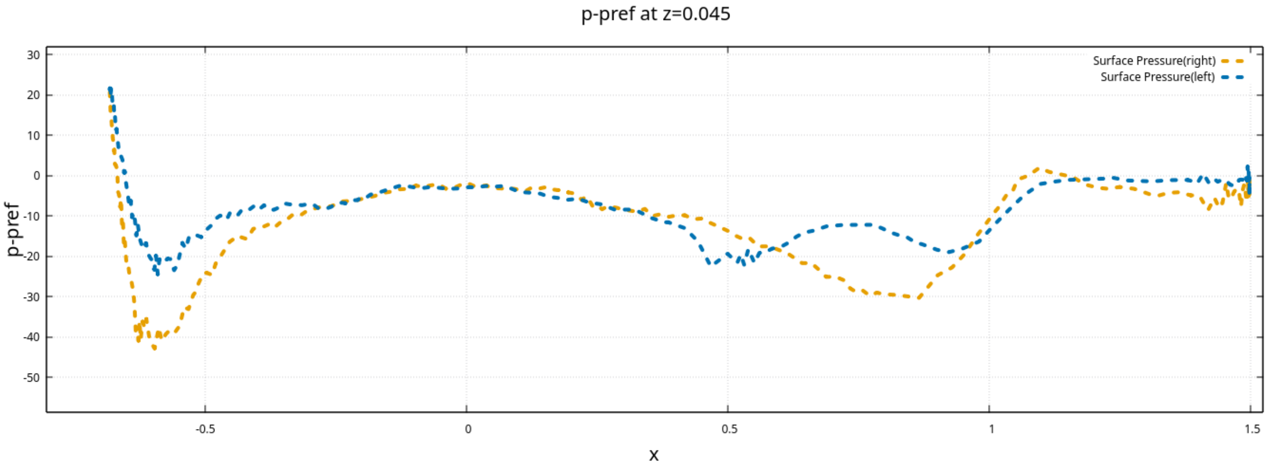


Figure 7.29 – Robin pressure at $z = 0.045$.

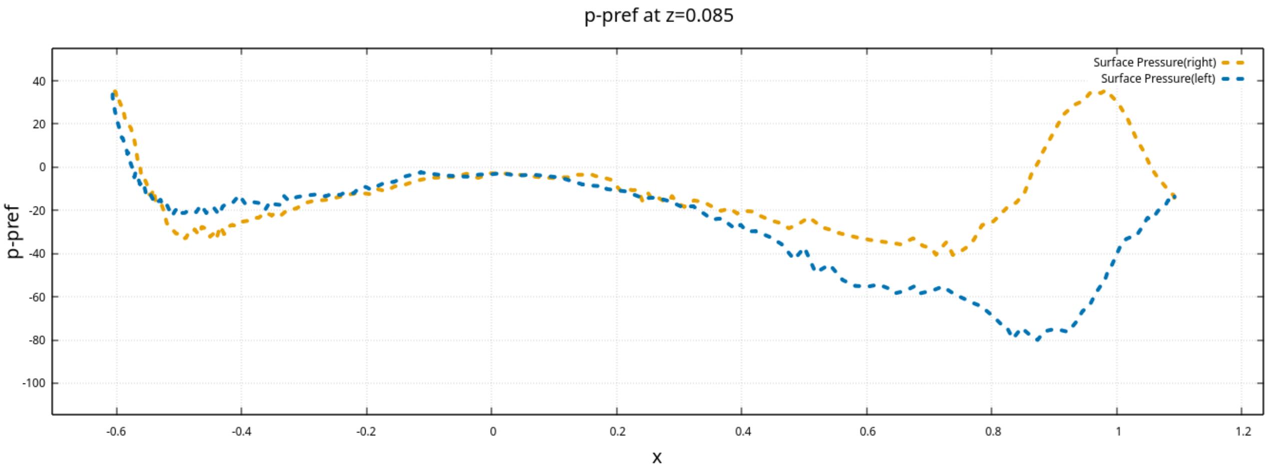


FIGURE 7.30 – Robin pressure at $z = 0.085$.

6 Conclusion

We have developed, in NiceFlow®, the MRF method for compressible flows. This method is not time consistent, but gives accurate predictions for steady situations. We have combined the MRF method with the anisotropic mesh adaptation. This has been first validated with the Caradonna-Tung rotor Norma test case. A second demonstrative computation concerns the Norma test case of a steady “hover” flow around the combination of the Caradonna-Tung rotor with the Robin fuselage.

CHAPITRE 8

A space and time fixed point mesh adaptation method

1 Introduction

In the context of CFD calculations, the mesh generation for accurate and robust numerical simulations of Reynolds-Averaged Navier-Stokes (RANS) equations is a time-consuming task. This results from the fact that meshes are traditionally considered as an input to the simulation pipeline. The common practices require a manual refinement of the computational domain and an adjustement of the timestep length both based on the *a priori* knowledge of the solution. This means that mesh generation and timestep length choice mainly rely on the experience and intuition of a skilled engineer to predict the flow and to adapt the discretization to the flow. It is evident that this operation can be manually accomplished only for simple geometries or academic flows where the solution is known. Whilst, this is not the case for complex geometries as well as for flow conditions exhibiting complicated features (e.g., shocks, supersonic shear layers, separation, etc.). In these cases, traditional "good practice guidelines" leads to burden the simulation pipeline or even turn out to be unfeasible.

Metric-based mesh adaptation is an efficient framework to generate adapted anisotropic meshes under time-delivering constraints, when a study at different physical conditions is demanded. It can be done in a feature-based mode, relying on minimizing the interpolation error of one or several sensors (=features) in L^p norm, or in a goal-oriented mode in which the error committed on a scalar output of a PDE is minimized with the use of an adjoint state. Both can be applied to steady calculations and to unsteady ones. We refer to [82, 83] for theoretical statements, and the monograph [39] for detailed descriptions of these methods. We consider the second case, that is the discretization of a PDE in the space-time domain $Q = \Omega \times [0, T]$. We advance in time and want to adapt the discretization to the solution. Two main options differ according to the spatial mesh adaptation.

(1) An important option is *mesh adaptation at each time step*, which consists in building at each time step a new adapted anisotropic mesh $\mathcal{H}(t)$ by defining a metric field $\mathcal{M}(t)$ on the computational domain Ω, which is optimal for this time level. This option may be expensive in terms of computational cost and may be of low accuracy due to errors committed in transferring solutions between too many successive meshes. Therefore this option is interesting from a theoretical point of view but it is of no practical use.

(2) A second strategy consists in *freezing the adapted mesh during several time steps.*

Then, it is mandatory that the mesh anticipates the flow behavior during these time steps, in other words during a given time subinterval. In [10, 12, 57] a transient fixed point (TFP) mesh adaptation algorithm has been proposed in order to master this issue. In the first version, the adaptation loop applies successively to each subinterval where the mesh is frozen. The error criterion is of *feature-based* type, measuring the interpolation error of a sensor chosen by the user. The TFP approach has been extended to a *goal-oriented* adaptation in [23] where several analyses were proposed for evaluating the convergence order of the TFP. In this context, the TFP was extended into the Global Transient Fixed Point (GTFP) in which the different meshes take into account a global space-time complexity evaluation and therefore are generated after the complete time resolution. This was mandatory as we had to solve the backward in time adjoint problem. A more complete accuracy and convergence analysis of GTFP, for a feature-based adaptation is proposed in [13] with many numerical examples.

In these works, either the time step is directly specified by the user, or the time step is assumed to be defined via a CFL stability condition related to an explicit time advancing. This is a useful option when an explicit time scheme is applied. In practice, it adapts quite well the time step to the solution as analyzed in [12]. Thanks to the CFL stability condition, the only unknown is the spatial mesh or the set of spatial meshes to perform the simulation.

The choice of a time step is defined in other terms when an implicit time advancing is used. Indeed, the size of the time step is no longer directly related to a stability condition. The time steps which are used can be notably larger than those permitted with an explicit time advancing. Large time steps induce a higher CPU efficiency, but the time approximation accuracy becomes an issue. Too large time steps degrade the prediction, too small time steps increase the computational cost. In this context, the time step (or time discretization) becomes also an unknown of the mesh adaptation problem.

Many attempts to control the time step size on an accuracy basis can be found in the literature. In [88], the two components of time error, namely truncation and implicit iterative errors are evaluated and controlled. In [87], the authors use an adjoint based output sensibility and a division of elements and time steps, addressing the largest error with the fewest additional space-time elements. A similar approach is combined with a space-time AMR in [70]. Another proposal for combining time adaptation with AMR is presented in [49]. Papers [91, 92] relies on an a posteriori analysis and error equidistribution in space and in time. A context closer to our is addressed in [36] where the authors adapt separately time and space. While giving interesting results, choosing a separate adaptation of the time discretization and the space discretization implies limitations in the global accuracy/efficiency compromise in the calculation. For example, a too fine mesh is useless if the time step is too large, a small time step is useless if local mesh size is too large.

The novelty of this work is to present a fully coupled approach where the temporal and the spatial errors are linked, hence the spatial error is impacting the time error and vice versa. Therefore, the goal is to obtain directly the space-time discretization which minimizes an error model under the constraint of a prescribed space-time discretization complexity. This strategy is developed in the framework of the Global Transient Fixed Point (GTFP) mesh adaptation, and extends GTFP to space-time adaptation in a direct manner. For the feature-based approach, the problem of the optimal simultaneous adaptation of the spatial mesh and the time step can be formulated under the form of a severely nonlinear optimization problem where the spatial and the temporal errors are tightly coupled. To solve it, we propose a slight simplification where both errors are loosely coupled in order to design a tractable accurate and efficient time step and mesh adaptive GTFP algorithm. In the case of a goal-oriented criterion, we demonstrate that the proposed formulation applies in a natural way.

In the present paper, the numerical study is restricted to the feature-based approach for which we demonstrate the validity of the new mesh and time step adaptation algorithms. Section 2 sets the notion of space-time continuous mesh which is mandatory to perform the theoretical analysis. Section 3 gives the considered space-time error models : first in the context of the 1D scalar advection and, second, in the context of the compressible Navier-Stokes equation for the feature-based and goal-oriented methods. The space-time error analysis providing the optimal spatial mesh and the optimal temporal mesh is carried out in the case of a space-time adaptation at each time step in Section 4 and in the case of a space-time adaptation for subintervals with the GTFP in Section 5. The Global Space-Time Transient Fixed Point algorithm is given in Section 6 and is applied to several flows with vortex shedding past a cylinder in Section 7. The paper ends, Section 8, with some concluding remarks and perspectives.

2 Space-time continuous mesh

The error analysis is based on the continuous mesh framework [82, 83] where the spatial domain discretization, i.e., the mesh, is represented by continuous functions, i.e., metric fields, defined on the computational domain. In this framework, the discrete error model is recast into a continuous one. Then, minimizing the continuous error model gives the optimal continuous mesh, solution of a continuous optimal system. The optimal adapted mesh is obtained by discretizing the continuous metric field, that is to say by generating a unit mesh with respect to the prescribed metric field [50, 85]. This section introduces the notion of space-time continuous mesh, its complexity, and the mechanics to generate the discrete space-time meshes.

A metric field $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is a 3×3 matrix field defined on the 3D spatial computational domain Ω, and such that for any $\mathbf{x}$ in Ω, $\mathcal{M}(\mathbf{x})$ is symmetric definite positive. A metric field $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is called (spatial) continuous mesh [82]. A spatial mesh represented by the metric field $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is any element of the class of meshes which are unit meshes for the metric field $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$.[1] The spatial complexity of a metric field :

$$\mathcal{C}_{\text{space}}(\mathcal{M}) = \int_\Omega \sqrt{det(\mathcal{M}(\mathbf{x}))}\, \mathrm{d}\mathbf{x}, \tag{8.1}$$

represents the number of vertices of the spatial discretization modelled by the metric field.

Definition 2.1 Space-time continuous mesh. *We call space-time continuous mesh* $(\mathcal{M}, \tau)$
the knowledge of the following ingredients :

(i) a time step function $\tau : t \in [0,T] \mapsto \tau(t) \in]0,T[$

(ii) for every $t \in [0,T]$ *a spatial metric field* $(\mathcal{M}(\mathbf{x},t))_{\mathbf{x}\in\Omega}$ *of spatial complexity* $\mathcal{N}(t) = \mathcal{C}_{\text{space}}(\mathcal{M}(t))$.

Remark 2.1 *Not all space-time continuous mesh allows to derive a space-time mesh. A space-time continuous mesh* $(\mathcal{M}, \tau)$ *is a valid parametrization of a space-time discretization if, for any* $t \in [0,T]$*, a unit mesh can be built from* $\mathcal{M}(t)$*, and if it exits an integer* $n_{step} \geq 1$ *such that the time step* τ *satisfies*

$$\int_0^T (\tau(t))^{-1}\mathrm{d}t \approx n_{step}.$$

Definition 2.2 Complexity. *The space-time complexity* $\mathcal{C}(\mathcal{M}, \tau)$ *of a space-time continuous mesh* $(\mathcal{M}, \tau)$ *is :*

$$\mathcal{C}_{st}(\mathcal{M}, \tau) \;=\; \int_0^T \mathcal{C}_{\text{space}}(\mathcal{M}(t))\; (\tau(t))^{-1}\mathrm{d}t. \tag{8.2}$$

From a given space-time continuous mesh, we can recover a fully-discrete time-advancing mesh by choosing a unit space-time mesh of it :

Definition 2.3 Unit space-time mesh. *Given a (valid and sufficiently smooth) space-time continuous mesh* $(\mathcal{M}, \tau)$*, a discrete space-time mesh* $(\mathcal{H}_k, t_k)_k$ *is unit with respect to* $(\mathcal{M}, \tau)$ *if it verifies :*

(i) since the time density $(\tau(t))^{-1}$ *is positive and satisfies* $\int_0^T (\tau(t))^{-1}\mathrm{d}t \approx n_{step}$ *we can successively build time levels* t_k *by putting :*

t_k *such that* $\int_{t_{k-1}}^{t_k} (\tau(t))^{-1}\mathrm{d}t = 1$,

stopping when it does not holds, for $n_{step} = integer\left(\int_0^T (\tau(t))^{-1}\mathrm{d}t\right)$.

---

[1] See [39]. In short, edges of the unit mesh are of length between $\sqrt{1/2}$ and $\sqrt{2}$ for the length induced by the metric $\mathcal{M}$.

(ii) At any time level t_k, $\mathcal{M}_k = \mathcal{M}(t_k)$ is used for generating a unit spatial mesh $\mathcal{H}_k$.

Remark 2.2 *Since each spatial mesh $\mathcal{H}_k$ generated has about $\mathcal{C}_{\text{space}}(\mathcal{M}_k)$ vertices, the total degrees of freedom in the time-advancing mesh $((\mathcal{H}_k)_k, (\tau_k)_k)$ is*

$$\mathcal{C}_{st}(\mathcal{M}, \tau) \; = \; \sum_{k=1}^{n_{step}} \mathcal{C}_{\text{space}}(\mathcal{M}_k).$$

3 Space-time error analysis

In the following, given a space-time continuous mesh, we provide a model of the approximation error estimating the error committed when using a unit discrete space-time mesh with respect to this space-time continuous mesh. First, such a model is proposed for the 1D scalar advection equation, then the case of the compressible Navier-Stokes equations is addressed.

3.1 Feature-based error model for the scalar advection equation

Let us consider the scalar advection model

$$u_t + cu_x = 0, \quad x \in \mathbb{R}.$$

We consider the usual continuous P_1 FEM approximation on a splitting of $\mathbb{R}$ in intervals $[x_i, x_{i+1}]$:

$$V_h = \left\{ \varphi_h \in \mathcal{C}^0(\mathbb{R}), \text{ supp}(\varphi_h) \text{ is compact, } \varphi_{h|_{[x_i,x_{i+1}]}} \text{ is affine} \right\},$$

$$u_h \in V_h, \quad \forall \varphi_h \in V_h, \;\; (\varphi_h, u_{h,t} + cu_{h,x}) = 0.$$

We choose the simplifying standpoint of a truncation error analysis of each separate term of the equation. We are first interested by the local spatial error :

$$\varepsilon_{\text{space}} = (\varphi_h, u_{h,t} + cu_{h,x} - (u_t + cu_x)) = 0.$$

A rough truncation error estimate writes this error in terms of mesh size Δx ($K_x \in \mathbb{R}$) :

$$|(\varphi_h, u_{h,t} + cu_{h,x} - (u_t + cu_x))| \leqslant K_x \, \Delta x (H_{u_t} + H_u) \Delta x,$$

where H_v holds for the absolute value of the Hessian of v. In the *continuous mesh framework*, if the 1D mesh $([x_i, x_{i+1}])_i$ is unit with respect to $(\mathcal{M}(x))_{x\in\Omega}$, then we have $\Delta x = \mathcal{M}(x)$. In other words, the continuous local error model writes :

$$|(\varphi_h, u_{h,t} + cu_{h,x} - (u_t + cu_x))| \leqslant K_x \,\mathrm{trace}\Big(\mathcal{M}^{-\frac{1}{2}}(H_{u_t} + H_u)\mathcal{M}^{-\frac{1}{2}}\Big).$$

with $K_x \in \mathbb{R}$ a constant depending on the dimension d. We also have to define, for all $t \in [0, T]$, a *time dependent time step function* $\Delta t = \tau(t)$, and a discrete time-derivative :

$$u_{h,\tau,t} \approx u_{h,t}\ ,$$

and we estimate the local time error ($K_t \in \mathbb{R}$) by :

$$|u_{h,\tau,t} - u_{h,t}| \leq K_t \tau^\alpha \left|\frac{\partial^{\alpha+1} u}{\partial t^{\alpha+1}}\right|.$$

If the second-order backward differencing formula is used, we have $\alpha = 2$ and $K_t = \frac{1}{3}$.

We can now define a *strongly coupled global space-time error model* which is based on the L^p-norm of the previous local time and space error models :

$$\overline{\mathcal{E}}^p_{\text{st}}(\mathcal{M}, \tau) = \int_0^T \int_\Omega \left[K_t\, \tau^\alpha \left|\frac{\partial^{\alpha+1} u}{\partial t^{\alpha+1}}\right| + K_x\, \Delta x (H_{u_t} + H_u)\Delta x\right]^p \mathrm{d}x\mathrm{d}t,$$

where Δx is the mesh size prescribed by $\mathcal{M}$. The analysis of $\overline{\mathcal{E}}^p_{\text{st}}$ is rather complex as the temporal and spatial errors are tightly coupled leading to a strongly non-linear model. We therefore propose a slight simplification where the temporal and spatial errors are loosely coupled in order to design a tractable error modeling. In the sequel, the following *loosely coupled global space-time error estimate* is analyzed :

$$\mathcal{E}^p_{\text{st}}(\mathcal{M}, \tau) = \int_0^T \int_\Omega \left[K_t \tau^\alpha \left|\frac{\partial^{\alpha+1} u}{\partial t^{\alpha+1}}\right|\right]^p + \Big[K_x \Delta x (H_{u_t} + H_u)\Delta x\Big]^p \mathrm{d}x\mathrm{d}t, \tag{8.3}$$

where we have a sum of a temporal error :

$$\mathcal{E}_{\text{time}}(\mathcal{M}, \tau) = \int_0^T K_t^p \tau^{\alpha p} \int_\Omega \left|\frac{\partial^{\alpha+1} u}{\partial t^{\alpha+1}}\right|^p \mathrm{d}x\mathrm{d}t,$$

and a spatial error :

$$\mathcal{E}_{\text{space}}(\mathcal{M}, \tau) = \int_0^T \int_\Omega \Big(K_x \Delta x (H_{u_t} + H_u)\Delta x\Big)^p \mathrm{d}x\mathrm{d}t.$$

3.2 Feature-based error model for CFD

Let $W = (\rho, \rho\mathbf{u}, \rho E)$ be the conservative variables vector where ρ denotes the density (kg/m^3), $\mathbf{u}$ the velocity (m/s), E the total energy per mass unit ($m^2 s^{-2}$). The compressible Navier-Stokes equations reads :

$$\Psi(W) = 0 \iff \left\{ \begin{array}{rcl} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) & = & 0\,, \\ \dfrac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p & - & \nabla \cdot \mathcal{T} = 0\,, \\ \dfrac{\partial (\rho E)}{\partial t} + \nabla \cdot ((\rho E + p)\mathbf{u}) & - & \nabla \cdot (\mathcal{T} \cdot \mathbf{u}) - \nabla \cdot (\lambda \nabla \theta) = 0\,, \\ +\text{Boundary conditions} & & \end{array} \right. \tag{8.4}$$

where p is the pressure (N/m^2), given by $p = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho \|\mathbf{u}\|^2\right)$, where γ is constant ($\gamma = 1.4$ in the sequel), θ the temperature (K) such that $\rho C_v \theta = E - \frac{1}{2}\rho\|\mathbf{u}\|^2$ (C_v being the specific heat at constant volume), μ the laminar dynamic viscosity ($kg/(ms)$) and λ the laminar conductivity. $\mathcal{T}$ is the laminar stress tensor :

$$\mathcal{T} = \mu \left[(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) - \frac{2}{3} \nabla .\, \mathbf{u}\, I \right] .$$

The variation of nondimensionalized laminar dynamic viscosity and conductivity coefficients μ and λ as function of the dimensional temperature T are defined by Sutherland's law :

$$\mu \;=\; \mu_\infty \left(\frac{\theta}{\theta_\infty}\right)^{\frac{3}{2}} \left(\frac{\theta_\infty + \text{Su}}{\theta + \text{Su}}\right) \quad \text{and} \quad \lambda \;=\; \lambda_\infty \left(\frac{\theta}{\theta_\infty}\right)^{\frac{3}{2}} \left(\frac{\theta_\infty + \text{Su}}{\theta + \text{Su}}\right) ,$$

where Su $= 110^\circ K$ is the Sutherland temperature and the index ∞ denotes reference quantities. The relation linking μ and λ is expressed from the Prandtl laminar number :

$$\text{Pr} = \frac{\mu C_p}{\lambda} \quad \text{with} \quad \text{Pr} = 0.72 \quad \text{for (dry) air}\,,$$

where C_p is the specific heat at constant pressure.

For the compressible Navier-Stokes equation, the feature-based (FB) error analysis leads to the following global space-time error model in L^p norm

$$\left(\mathcal{E}^{\text{FB}}_{\text{st}}\right)^p(\mathcal{M}, \tau) \;=\; \mathcal{E}^{\text{FB}}_{\text{time}}(\mathcal{M}, \tau) \;+\; \mathcal{E}^{\text{FB}}_{\text{space}}(\mathcal{M}, \tau), \tag{8.5}$$

with

$$\mathcal{E}^{\text{FB}}_{\text{time}}(\mathcal{M},\tau) = \int_0^T K_t^p \tau^{\alpha p} \int_\Omega \left| \frac{\partial^{\alpha+1} W}{\partial t^{\alpha+1}} \right|^p \mathrm{d}\mathbf{x}\mathrm{d}t, \tag{8.6}$$

and

$$\mathcal{E}^{\text{FB}}_{\text{space}}(\mathcal{M},\tau) = \int_0^T \int_\Omega \left(\text{trace} \left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\, \mathbf{H}(\mathbf{x},t)\, \mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t) \right) \right)^p \mathrm{d}\mathbf{x}\mathrm{d}t, \tag{8.7}$$

where $\mathbf{H} = |H_{u_t} + H_u|$ depends on sensor u computed from W the solution of the state Equation (8.4).

3.3 Goal-oriented error model for CFD

We consider the goal-oriented formulation as introduced in [9, 23] and keep the notations of these papers. The goal is to minimize the error $(g, W - W_h)$ committed in the approximation of the functional (or scalar output) :

$$j = (g, W),$$

where W is the exact solution of the state Equation (8.4) and W_h the approximate solution. The novelty with respect to [9, 23] is that the error on the time discretization is also taken into account. This leads to an extra term $\mathcal{E}^{\text{GO}}_{\text{time}}(\mathcal{M},\tau)$ in the goal-oriented (GO) global space-time error estimate :

$$|(g, W_h - W)| \;\approx\; \mathcal{E}^{\text{GO}}_{\text{st}}(\mathcal{M},\tau) \;=\; \mathcal{E}^{\text{GO}}_{\text{space}}(\mathcal{M},\tau) + \mathcal{E}^{\text{GO}}_{\text{time}}(\mathcal{M},\tau). \tag{8.8}$$

We remind that the goal-oriented error estimate is in L^1 norm thus, here, we have $p = 1$. The global temporal error model is

$$\mathcal{E}^{\text{GO}}_{\text{time}}(\mathcal{M},\tau) = \int_0^T \int_\Omega K_t\, \tau^\alpha \left| W^* \, \frac{\partial^{\alpha+1} W}{\partial t^{\alpha+1}} \right| \mathrm{d}\mathbf{x}\, \mathrm{d}t, \tag{8.9}$$

where W^* is the adjoint state, solution of the adjoint system :

$$-W_t^* + \left(\frac{\partial \Psi}{\partial W} \right)^* W^* = g.$$

We recall the spatial goal-oriented error estimate developed and progressively enriched in [9, 22, 23, 84]. The goal-oriented error model $\mathcal{E}^{\text{GO}}_{\text{space}}(\mathcal{M},\tau)$ is expressed in terms of

Euler fluxes $\mathcal{F}^E$, viscous fluxes $\mathcal{F}^V$, and boundary Euler fluxes $\bar{\mathcal{F}}$:

$$\begin{aligned}\mathcal{E}^{\text{GO}}_{\text{space}}(\mathcal{M},\tau) \approx & \int_0^T \int_\Omega |W_t^*| \; |W-\pi_{\mathcal{M}}W| \,\mathrm{d}\mathbf{x}\,\mathrm{d}t \\ &+ \int_0^T \int_\Omega |\frac{\partial \mathcal{F}^E}{\partial W}\,\nabla W^*| \; |W-\pi_{\mathcal{M}}W| \,\mathrm{d}\mathbf{x}\,\mathrm{d}t \\ &+ \int_0^T \int_\Omega |\frac{\partial \mathcal{F}^V}{\partial \nabla W}\,H(W^*)| \; |W-\pi_{\mathcal{M}}W| \,\mathrm{d}\mathbf{x}\,\mathrm{d}t \\ &+ \int_0^T \int_\Gamma |W^*| \; |(\bar{\mathcal{F}}(W)-\pi_{\mathcal{M}}\bar{\mathcal{F}}(W))\cdot\mathbf{n}| \,\mathrm{d}\Gamma\,\mathrm{d}t \,.\end{aligned}$$

Neglecting the boundary term, we note that all the terms have the form of a weighted interpolation error in L^1 norm on the conservative variables. We therefore deduce the following global spatial error model :

$$\mathcal{E}^{\text{GO}}_{\text{space}}(\mathcal{M},\tau) \;\approx\; \int_0^T \int_\Omega \text{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\,\mathbf{H}(\mathbf{x},t)\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\right)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\,, \tag{8.10}$$

$$\text{with} \quad \mathbf{H}(\mathbf{x},t) = \left|W_t^* + \frac{\partial \mathcal{F}^E}{\partial W}\,\nabla W^* + \frac{\partial \mathcal{F}^V}{\partial \nabla W}\,H(W^*)\right| \; |H(W)|\,,$$

where $H(W)$ (resp. $H(W^*)$) is the Hessian of W (resp. W^*).

3.4 Unified error model for CFD

By analyzing the feature-based error model (8.5,8.6,8.7) and the goal-oriented error model (9.35,8.9,8.10), we note that their formulations are similar. Consequently, both models can be unified as follows :

$$\mathcal{E}^p_{\text{st}}(\mathcal{M},\tau) = \mathcal{E}_{\text{time}}(\mathcal{M},\tau) + \mathcal{E}_{\text{space}}(\mathcal{M},\tau), \tag{8.11}$$

with

$$\mathcal{E}_{\text{time}}(\mathcal{M},\tau) = \int_0^T K_t^p\,\tau^{\alpha p} \int_\Omega \left|W^*\frac{\partial^{\alpha+1}W}{\partial t^{\alpha+1}}\right|^p \mathrm{d}\mathbf{x}\,\mathrm{d}t, \tag{8.12}$$

and

$$\mathcal{E}_{\text{space}}(\mathcal{M},\tau) = \int_0^T \int_\Omega \left(\text{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\,\mathbf{H}(\mathbf{x},t)\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\right)\right)^p \mathrm{d}\mathbf{x}\mathrm{d}t\,, \tag{8.13}$$

where :

- in the feature-based case : a sensor u is computed from W, allowing to replace $W^*\dfrac{\partial^{\alpha+1}W}{\partial t^{\alpha+1}}$ by a time derivative $\dfrac{\partial^{\alpha+1}u}{\partial t^{\alpha+1}}$ of the sensor, and setting $\mathbf{H} = \left|H_{u_t} + H_u\right|$.
- in the goal-oriented case : $p = 1$, W^* is the adjoint state, and $\mathbf{H}$ is defined as in Equation (8.10).

4 Analysis for all-time adaptation

This section defines an optimal space-time adaptive strategy when the mesh is adapted at each time step. Let $\mathcal{N}_{\text{st}}$ be an integer prescribed by the user which represents the space-time mesh complexity, we call *mesh adaptation problem* the following problem :

Find $(\mathcal{M},\tau)$ *which minimizes* $\mathcal{E}_{\text{st}}(\mathcal{M},\tau)$ *under the constraint* $\mathcal{C}_{\text{st}}(\mathcal{M},\tau)=\mathcal{N}_{\text{st}}$.

4.1 Analysis for all-time mesh adaptation at a given time

At a fixed time t, given a spatial mesh complexity $\mathcal{N}(t)$, we know the optimal metric $\mathcal{M}_{\text{opt}}(\mathbf{x},t)$ under the constraint $\mathcal{C}_{\text{space}}(\mathcal{M}(t))=\mathcal{N}(t)$ by minimizing the spatial part of the error model in L^p norm :

$$\mathcal{M}_{\text{opt}}(t)=\underset{\mathcal{M}}{\text{Arg}\min}\int_\Omega \left(\text{trace}\left((\mathcal{M}(\mathbf{x},t))^{-\frac{1}{2}}\mathbf{H}(\mathbf{x},t)(\mathcal{M}(\mathbf{x},t))^{-\frac{1}{2}}\right)\right)^p \,\mathrm{d}\mathbf{x}.$$

The pointwise optimal metric is given by [82] :

$$\mathcal{M}_{\text{opt}}(\mathbf{x},t)=\mathcal{N}(t)^{\frac{2}{d}}\left(\int_\Omega(\det\mathbf{H}(\mathbf{x},t))^{\frac{p}{2p+d}}\,\mathrm{d}\mathbf{x}\right)^{-\frac{2}{d}}(\det\mathbf{H}(\mathbf{x},t))^{-\frac{1}{2p+d}}\,\mathbf{H}(\mathbf{x},t), \tag{8.14}$$

and the related optimal spatial error is :

$$\mathcal{E}_{\text{space}}(\mathcal{M}_{\text{opt}},t)=d^pK_x^p\,\mathcal{N}(t)^{-\frac{2p}{d}}\left(\int_\Omega(\det\mathbf{H}(\mathbf{x},t))^{\frac{p}{2p+d}}\,\mathrm{d}\mathbf{x}\right)^{\frac{2p+d}{d}}, \tag{8.15}$$

where d is the spatial domain dimension, and $K_x=\frac{1}{20}$ if $d=3$ or $K_x=\frac{1}{8}$ if $d=2$ (see [39], Corollary 1 of Theorem 4.2.2.).

4.2 Analysis for all-time adaptation over time interval

We are now interested in minimizing the space-time error over the time interval $[0,T]$. The unified space-time error model (9.38,8.12,8.13) becomes by using the optimal spatial error term (8.14) :

$$\mathcal{E}_{\text{st}}^p(\mathcal{M},\tau)=\int_0^T\left[K_t^p\,\tau(t)^{\alpha p}\int_\Omega\left|W^*(\mathbf{x},t)\frac{\partial^{\alpha+1}W(\mathbf{x},t)}{\partial t^{\alpha+1}}\right|^p\mathrm{d}\mathbf{x}+d^pK_x^p\,\mathcal{N}(t)^{-\frac{2p}{d}}\left(\int_\Omega(\det\mathbf{H}(\mathbf{x},t))^{\frac{p}{2p+d}}\,\mathrm{d}\mathbf{x}\right)^{\frac{2p+d}{d}}\right]$$

under the space-time complexity constraint

$$\mathcal{C}_{\text{st}}(\mathcal{M},\tau) \;=\; \int_0^T \mathcal{N}(t)\,(\tau(t))^{-1}\mathrm{d}t \;=\; \mathcal{N}_{\text{st}}.$$

Analyzing the formulations of $\mathcal{E}^p_{\text{st}}(\mathcal{M},\tau)$ and $\mathcal{C}_{\text{st}}(\mathcal{M},\tau)$, we note that the error model and the constraint are expressed in terms of $\mathcal{N}(t)$ and τ but no more in term of $\mathcal{M}$. Consequently, we reformulate the optimization problem in terms of $\mathcal{N}(t)$ and τ by introducing two functions :

$$\begin{aligned}
&\mathfrak{F} : \mathcal{N} \in \mathcal{C}^0[0,T;\mathbb{R}] \mapsto \mathfrak{F}(\mathcal{N}) \in \mathcal{C}^0[0,T;\mathbb{R}],\\
&\overline{\mathcal{G}} : (\mathcal{N},\tau) \in \left(\mathcal{C}^0[0,T;\mathbb{R}]\right)^2 \mapsto \overline{\mathcal{G}}(\mathcal{N},\tau) \in \mathcal{C}^0[0,T;\mathbb{R}],
\end{aligned}$$

such that

$$\begin{aligned}
&\overline{\mathcal{G}}(\mathcal{N},\tau) = K_t^p\,\tau(t)^{\alpha p}\int_\Omega \left| W^*(\mathbf{x},t)\frac{\partial^{\alpha+1}W(\mathbf{x},t)}{\partial t^{\alpha+1}}\right|^p \mathrm{d}\mathbf{x}\,,\\
&\mathfrak{F}(\mathcal{N}) = d^p K_x^p \mathcal{N}(t)^{-\frac{2p}{d}}\left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{p}{2p+d}}\ \mathrm{d}\mathbf{x}\right)^{\frac{2p+d}{d}}\ .
\end{aligned}$$

Then, the space-time mesh adaptation problem becomes :

$$\left\{\begin{aligned}
&\text{Find } (\mathcal{N}_{\text{opt}},\tau_{\text{opt}}) = \underset{\mathcal{N},\tau}{\text{Arg min}} \int_0^T \left(\overline{\mathcal{G}}(\mathcal{N},\tau) + \mathfrak{F}(\mathcal{N})\right)\,\mathrm{d}t\,,\\
&\text{such that } \mathcal{C}_{\text{st}}(\mathcal{N},\tau) = \int_0^T \mathcal{N}(t)\,(\tau(t))^{-1}\,\mathrm{d}t = \mathcal{N}_{\text{st}}.
\end{aligned}\right.$$

Let us start with a change of variables to avoid nonlinear constraints, namely $(\mathcal{N},\,\zeta) = (\mathcal{N},\,\mathcal{N}\tau^{-1})$,

$$\mathcal{G}(\mathcal{N},\zeta) = K_t^p\,\zeta(t)^{-\alpha p}\mathcal{N}(t)^{\alpha p}\int_\Omega \left| W^*(\mathbf{x},t)\frac{\partial^{\alpha+1}W(\mathbf{x},t)}{\partial t^{\alpha+1}}\right|^p \mathrm{d}\mathbf{x}\,,$$

so the constraint becomes $\int_0^T \zeta(t)\,\mathrm{d}t = \mathcal{N}_{\text{st}}$. Now, by setting

$$\begin{aligned}
&\mathcal{U}(t) = \alpha p\,K_t^p\,\int_\Omega \left| W^*(\mathbf{x},t)\frac{\partial^{\alpha+1}W(\mathbf{x},t)}{\partial t^{\alpha+1}}\right|^p \mathrm{d}\mathbf{x}\,,\\
&\mathcal{K}(t) = 2p\,d^{p-1}K_x^p\left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{p}{2p+d}}\ \mathrm{d}\mathbf{x}\right)^{\frac{2p+d}{d}}\ ,
\end{aligned}$$

the derivatives of functions $\mathfrak{F}$ and $\mathcal{G}$ are given by

$$\begin{aligned}
\frac{\partial\mathfrak{F}}{\partial\mathcal{N}}\,\delta\mathcal{N} \;&=\; -\mathcal{N}^{-\frac{2p+d}{d}}\mathcal{K}\,\delta\mathcal{N},\\
\frac{\partial\mathcal{G}}{\partial\mathcal{N}}\,\delta\mathcal{N} \;&=\; \zeta^{-\alpha p}\mathcal{N}^{\alpha p-1}\mathcal{U}\,\delta\mathcal{N},\\
\frac{\partial\mathcal{G}}{\partial\zeta}\,\delta\zeta \;&=\; -\zeta^{-\alpha p-1}\mathcal{N}^{\alpha p}\mathcal{U}\,\delta\zeta.
\end{aligned}\tag{8.16}$$

The optimal condition writes

$$\begin{cases} \displaystyle\int_0^T \left(\frac{\partial \mathfrak{F}}{\partial \mathcal{N}} + \frac{\partial \mathcal{G}}{\partial \mathcal{N}} \right) \delta\mathcal{N} \, \mathrm{d}t = 0 \,, \quad \forall\, \delta\mathcal{N}, \\ \displaystyle\int_0^T \frac{\partial \mathcal{G}}{\partial \zeta} \, \delta\zeta \, \mathrm{d}t = 0 \,, \quad \forall\, \delta\zeta \text{ such that } \int_0^T \delta\zeta \, \mathrm{d}t = 0 \,, \end{cases}$$

from which we deduce

$$\begin{cases} \dfrac{\partial \mathfrak{F}}{\partial \mathcal{N}} + \dfrac{\partial \mathcal{G}}{\partial \mathcal{N}} = 0, \\ \dfrac{\partial \mathcal{G}}{\partial \zeta} = -C, \end{cases}$$

where, according to Relations (8.16), C is a positive constant not depending in time. The second equation writes

$$\zeta(t)^{-\alpha p - 1} \mathcal{N}(t)^{\alpha p} \mathcal{U}(t) = C,$$

and gives

$$\mathcal{N}(t) = \left(\frac{C}{\mathcal{U}(t)} \right)^{\frac{1}{\alpha p}} \zeta(t)^{\frac{\alpha p + 1}{\alpha p}} \qquad \text{or} \qquad \zeta(t) = \left(\frac{\mathcal{U}(t)}{C} \right)^{\frac{1}{\alpha p + 1}} \mathcal{N}(t)^{\frac{\alpha p}{\alpha p + 1}} \,. \tag{8.17}$$

Thanks to Equation (8.17), the first equation gives

$$\mathcal{N}(t) = C^{-\frac{\alpha d}{\lambda}} \mathcal{K}(t)^{\frac{d(\alpha p + 1)}{p \lambda}} \mathcal{U}(t)^{-\frac{d}{p\lambda}}, \quad \text{with} \quad \lambda = 2(\alpha p + 1) + \alpha d \,.$$

Recalling that $\zeta = \mathcal{N} \tau^{-1}$ we have

$$\tau(t) = \left(\frac{C}{\mathcal{U}(t)} \right)^{\frac{1}{\alpha p + 1}} \mathcal{N}(t)^{\frac{1}{\alpha p + 1}},$$

and the constraint on the complexity gives

$$\mathcal{N}_{\text{st}} = \int_0^T \mathcal{N}(t) \tau(t)^{-1} \, \mathrm{d}t \; = \; C^{-\frac{\alpha d + 2}{\lambda}} \int_0^T \mathcal{K}(t)^{\frac{\alpha d}{\lambda}} \mathcal{U}(t)^{\frac{2}{\lambda}} \, \mathrm{d}t \,,$$

from which we deduce

$$C = \mathcal{N}_{\text{st}}^{-\frac{\lambda}{\alpha d + 2}} \left(\int_0^T \mathcal{K}(t)^{\frac{\alpha d}{\lambda}} \mathcal{U}(t)^{\frac{2}{\lambda}} \, \mathrm{d}t \right)^{\frac{\lambda}{\alpha d + 2}} \,.$$

For clarity, we introduce the global variable $\mathcal{S} = \int_0^T \mathcal{K}(t)^{\frac{\alpha d}{\lambda}} \mathcal{U}(t)^{\frac{2}{\lambda}} \,\mathrm{d}t$ such that $C = \mathcal{N}_{\text{st}}^{-\frac{\lambda}{\alpha d+2}} \mathcal{S}^{\frac{\lambda}{\alpha d+2}}$. Finally, using the value of the constant C, the solution of the optimization problem is given by

$$\begin{aligned} \mathcal{N}_{\text{opt}}(t) &= \mathcal{N}_{\text{st}}^{\frac{\alpha d}{\alpha d+2}}\ \mathcal{S}^{-\frac{\alpha d}{\alpha d+2}}\ \mathcal{K}(t)^{\frac{d(\alpha p+1)}{p\lambda}}\ \mathcal{U}(t)^{-\frac{d}{p\lambda}}\,, \\ \tau_{\text{opt}}(t) &= \mathcal{N}_{\text{st}}^{-\frac{2}{\alpha d+2}}\ \mathcal{S}^{\frac{2}{\alpha d+2}}\ \mathcal{K}(t)^{\frac{d}{p\lambda}}\ \mathcal{U}(t)^{-\frac{2p+d}{p\lambda}}\,, \end{aligned}$$

The optimal spatial continuous mesh at each time step that minimizes the space-time error is simply obtained by using the above solution in Formula (8.14) :

$$\mathcal{M}_{\text{opt}}(\mathbf{x},t) = \mathcal{N}_{\text{opt}}(t)^{\frac{2}{d}} \left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{p}{2p+d}}\ \mathrm{d}\mathbf{x} \right)^{-\frac{2}{d}} (\det \mathbf{H}(\mathbf{x},t))^{-\frac{1}{2p+d}}\, \mathbf{H}(\mathbf{x},t)\,.$$

This results shows that the space-time error model optimally distributes the number of vertices at each time step in order to minimize the space-time error.

Proposition 4.1 (Optimal space-time continuous mesh at each time step)
We consider the global space-time error model in L^p norm given by Equation (9.38). We define the two time dependent functions $\mathcal{U}$ and $\mathcal{K}$:

$$\begin{aligned} \mathcal{U}(t) &= \alpha p\, K_t^p \int_\Omega \left| W^*(\mathbf{x},t) \frac{\partial^{\alpha+1} W(\mathbf{x},t)}{\partial t^{\alpha+1}} \right|^p \mathrm{d}\mathbf{x}\,, \\ \mathcal{K}(t) &= 2p\, d^{p-1} K_x^p \left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{p}{2p+d}}\ \mathrm{d}\mathbf{x} \right)^{\frac{2p+d}{d}}\,. \end{aligned}$$

K_t and α depends on the chosen implicit time integration scheme, for instance for the second-order backward differencing formula we have $\alpha = 2$ and $K_t = \frac{1}{3}$. K_x depends on the spatial domain dimension d with $K_x = \frac{1}{8}$ in 2D and $K_x = \frac{1}{20}$ in 3D. We also introduce the following global variable :

$$\mathcal{S} \;=\; \int_0^T \mathcal{K}(t)^{\frac{\alpha d}{\lambda}} \mathcal{U}(t)^{\frac{2}{\lambda}}\, \mathrm{d}t\,.$$

Then, for any time t, the optimal spatial mesh complexity $\mathcal{N}_{\text{opt}}(t)$ and the optimal time step $\tau_{\text{opt}}(t)$ are given by :

$$\begin{aligned} \mathcal{N}_{\text{opt}}(t) &= \mathcal{N}_{st}^{\frac{\alpha d}{\alpha d+2}}\ \mathcal{S}^{-\frac{\alpha d}{\alpha d+2}}\ \mathcal{K}(t)^{\frac{d(\alpha p+1)}{p\lambda}}\ \mathcal{U}(t)^{-\frac{d}{p\lambda}}\,, \\ \tau_{\text{opt}}(t) &= \mathcal{N}_{st}^{-\frac{2}{\alpha d+2}}\ \mathcal{S}^{\frac{2}{\alpha d+2}}\ \mathcal{K}(t)^{\frac{d}{p\lambda}}\ \mathcal{U}(t)^{-\frac{2p+d}{p\lambda}}\,, \end{aligned}$$

with $\lambda = 2(\alpha p + 1) + \alpha d$, and the optimal spatial continuous mesh is :

$$\mathcal{M}_{\text{opt}}(\mathbf{x},t) = \mathcal{N}_{\text{opt}}(t)^{\frac{2}{d}} \left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{p}{2p+d}}\ \mathrm{d}\mathbf{x} \right)^{-\frac{2}{d}} (\det \mathbf{H}(\mathbf{x},t))^{-\frac{1}{2p+d}}\, \mathbf{H}(\mathbf{x},t)\,,$$

where for the feature-based case $\mathbf{H} = \left| H_{u_t} + H_u \right|$, and for the goal-oriented case $\mathbf{H}$ is given by Relation (8.10) and $p = 1$.

5 Analysis with time subintervals

The analysis of last section considers that the spatial mesh is updated at each time step. As remarked in previous works, this option has two important disadvantages, namely (i) the computational effort consumed in regenerating the mesh at each time step is generally prohibitive for practical applications, and (ii) the loss of accuracy in transferring the solution from one spatial mesh to the next one is also generally too large.

5.1 The Global Transient Fixed Point (GTFP) algorithm

To avoid these disadvantages, the Transient Fixed Point (TFP) was introduced in [10, 11]. The idea was to keep the same adapted mesh for a given time subinterval where the flow solver performs several time steps. In this first version, the mesh adaptation loop applies successively to each subinterval. As a result, this approach does not allow an analysis of the space-time error because the error estimation is done subinterval by subinterval. Therefore, we cannot optimize optimize the complete space-time mesh. To solve this issue, the Global Transient Fixed Point (GTFP) was proposed in [13, 23] in which a global space-time error evaluation is done after the complete time resolution of the simulation. This section recalls its main features.

The simulation time frame $[0, T[$ is split into n_{adap} subintervals of same length (Figure 8.1) :

$$[0,T[= \ [0=t_0,t_1[\cup \ldots \cup [t_{i-1},t_i[\cup \ldots \cup [t_{n_{adap}-1},t_{n_{adap}} = T[= \bigcup_{i=1}^{n_{adap}} [t_{i-1},t_i[.$$

The number n_{adap} *of adaptation time subintervals* $[t_{i-1}, t_i[$ is a discretization parameter to be specified by the user. Each subinterval $[t_{i-1}, t_i[$ contains a large number of time steps of the flow solver. The GTFP algorithm is schematized in Algorithm 8.1 where $\mathcal{H}$, $\mathcal{S}$ and $\mathcal{M}$

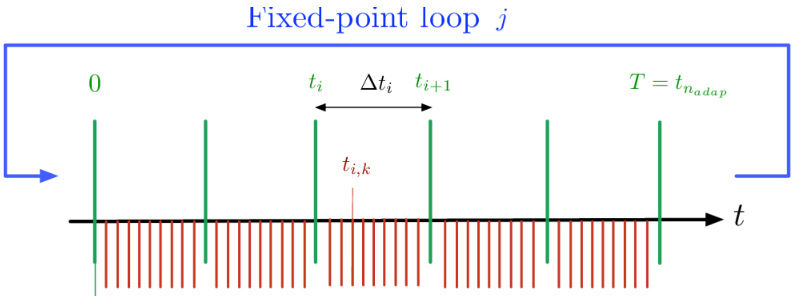


FIGURE 8.1 – Time splitting of the GTFP mesh adaptation algorithm. Subintervals (in green) used for the transient process and time steps (in red).

denote respectively meshes, solutions and metrics. *The external loop* applies a fixed point on the mesh adaptation process to converge the non-linear space-time mesh adaptation problem. *In the internal loop*, knowing the spatial mesh, the time subinterval $[t_{i-1}, t_i[$ is divided into m time-integration intervals $[t_i^k, t_i^{k+1}[$, with $k = 0, ..., m$ and $t_i^0 = t_{i-1}$, , $t_i^m = t_i$. For t_i^0 the solution is given by interpolation from the previous mesh, and for any $k = 0, ..., m-1$, the flow variables are advanced from time level t_i^k to time level t_i^{k+1} by means of the numerical scheme. *Before the end of the external loop*, a global error analysis produces the complete series of metrics and meshes for the next computation on the n_{adap} subintervals.

Algorithm 8.1 – GTFP : Global Transient Fixed Point for Unsteady Flows [13]

```
Initial mesh and solution $(\mathcal{H}_0, \mathcal{S}_0^0)$ and set targeted space-time complexity $\mathcal{N}_{\text{st}}$
# Fixed-point loop to converge the global space-time mesh adaptation problem
For $j = 1, n_{ptfx}$
  # Adaptive loop to advance the solution in time on time frame $[0, T[$
  1. For $i = 1, n_{adap}$      # Advance the solution in time in subinterval $[t_{i-1}, t_i[$
       (a) $\mathcal{S}_{0,i}^j$ = Interpolate conservatively next subinterval initial solution from $(\mathcal{H}_{i-1}^j, \mathcal{S}_{i-1}^j, \mathcal{H}_i^j)$ ;
       (b) $\mathcal{S}_i^j$ = Compute solution on subinterval from pair $(\mathcal{H}_i^j, \mathcal{S}_{0,i}^j)$ ;
       (c) $|\mathbf{H}|_i^j$ = Compute subinterval Hessian-metric from solution sample $(\mathcal{H}_i^j, \{\mathcal{S}_i^j(k)\}_{k=1,nk})$ ;
     EndFor
  2. $\mathcal{C}^j$ = Compute space-time complexity from all Hessian-metrics $(\{|\mathbf{H}|_i^j\}_{i=1,n_{adap}})$ ;
  3. $\{\mathbf{M}_i^j\}_{i=1,n_{adap}}$ = Compute all subinterval unsteady metrics $(\mathcal{C}^j, \{|\mathbf{H}|_i^j\}_{i=1,n_{adap}})$ ;
  4. $\{\mathbf{M}_i^j\}_{i=1,n_{adap}}$ = Metric gradation on all subinterval unsteady metrics $\{\mathbf{M}_i^j\}_{i=1,n_{adap}}$ ;
  5. $\{\mathcal{H}_i^{j+1}\}_{i=1,n_{adap}}$ = Generate all subinterval adapted meshes $(\{\mathcal{H}_i^j,\ \mathbf{M}_i^j\}_{i=1,n_{adap}})$ ;
EndFor
```

Several analyses of the GTFP are proposed in [23] and [39] (Lemma 2.9.3). In particular, for the 3D anisotropic case, a necessary condition for second order *spatial* convergence is that subintervals $[t_{i-1}, t_i[$ are **two times** smaller for a four times spatial smaller error [23]. As concerns *space-time* convergence in $\mathbf{L}^1([0, T[; \mathbf{L}^1(\Omega))$, which is the convergence in terms of the space-time complexity $\mathcal{N}_{\text{st}}$, it can be obtained in many cases at order $8/5$ by dividing the time step by a factor $4 : \tau \to \tau/4$ and passing from $(\mathcal{N}_{\text{st}}, n_{adap})$ to $(32\mathcal{N}_{\text{st}}, 2n_{adap})$ [23]. Further estimates are proposed in [13].

The present work proposes the extension of the GTFP to time-accurate implicit schemes. The central novelty is the *simultaneous adaptation of the mesh and the time steps*. It relies on the definition of a time-advancing mesh.

5.2 Notations for space-time Global Transient Fixed Point

It is useful to specify how the main continuous mesh variables are depending on the time variable t. In the context of the GTFP, we have :

1. the mesh complexity $\mathcal{N}(t)$ is a constant over each subinterval :

$$\forall\, i = 1, ..., n_{adap},\ \ \forall\, t \in [t_{i-1}, t_i[,\ \ \mathcal{N}(t) = \mathcal{N}^i \in \mathbb{R}, \tag{8.18}$$

2. the continuous mesh (metric field) $(\mathcal{M}(\mathbf{x}, t))_{\mathbf{x}\in\Omega}$ is constant over each subinterval :

$$\forall\, i = 1, ..., n_{adap},\ \ \forall\, t \in [t_{i-1}, t_i[,\ \ (\mathcal{M}(\mathbf{x}, t))_{\mathbf{x}\in\Omega} = (\mathcal{M}^i(\mathbf{x}))_{\mathbf{x}\in\Omega}, \tag{8.19}$$

3. the associated adapted mesh is also fixed over each subinterval :

$$\forall\, i = 1, ..., n_{adap},\ \ \forall\, t \in [t_{i-1}, t_i[,\ \ \mathcal{H}(t) = \mathcal{H}^i. \tag{8.20}$$

The complexities, continuous meshes and adapted meshes are solely changing when passing from subinterval $[t_{i-1}, t_i[$ to the next one $[t_i, t_{i+1}[$.

This section presents the optimal choice satisfying the above particular properties by solving the related optimization problem. We are again interested in minimizing the *loosely coupled space-time error model* :

$$\mathcal{E}^p_{\text{st}}(\mathcal{M}, \tau) = \int_0^T \left[\int_\Omega \left(K_t\, \tau(t)^\alpha \left| W^*(\mathbf{x}, t) \frac{\partial^{\alpha+1} W(\mathbf{x}, t)}{\partial t^{\alpha+1}} \right| \right)^p \mathrm{d}\mathbf{x} + \int_\Omega \left(\text{trace} \left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x}, t) \mathbf{H}(\mathbf{x}, t) \mathcal{M}^{-\frac{1}{2}}(\mathbf{x}, t) \right) \right)^p$$

under the space-time complexity constraint

$$\mathcal{C}_{\text{st}}(\mathcal{M}, \tau) \ = \ \int_0^T \mathcal{N}(t) (\tau(t))^{-1}\, \mathrm{d}t \ = \ \mathcal{N}_{\text{st}}\,.$$

As previously, we denote the first term of our error model $\mathcal{E}_{\text{time}}(\mathcal{M}, \tau)$ and the second term $\mathcal{E}_{\text{space}}(\mathcal{M}, \tau)$. We then rewrite each component of the error model in terms of subintervals using the above notations :

$$\mathcal{E}_{\text{time}}(\mathcal{M}, \tau) = \sum_{i=1}^{n_{adap}} \mathcal{E}^i_{\text{time}}(\mathcal{M}^i, \tau) = \sum_{i=1}^{n_{adap}} \int_{t_{i-1}}^{t_i} K_t^p\, \tau(t)^{\alpha p} \left(\int_\Omega \left| W^*(\mathbf{x}, t) \frac{\partial^{\alpha+1} W(\mathbf{x}, t)}{\partial t^{\alpha+1}} \right|^p \mathrm{d}\mathbf{x} \right) \mathrm{d}t,$$

$$\mathcal{E}_{\text{space}}(\mathcal{M}, \tau) = \sum_{i=1}^{n_{adap}} \mathcal{E}^i_{\text{space}}(\mathcal{M}^i, \tau) = \sum_{i=1}^{n_{adap}} \int_{t_{i-1}}^{t_i} \int_\Omega \left(\text{trace} \left((\mathcal{M}^i(\mathbf{x}))^{-\frac{1}{2}} \mathbf{H}(\mathbf{x}, t) (\mathcal{M}^i(\mathbf{x}))^{-\frac{1}{2}} \right) \right)^p \mathrm{d}\mathbf{x}\mathrm{d}t.$$

This defines our space-time error model for a GTFP mesh adaptation with n_{adap} subintervals.

5.3 Spatial mesh optimization on a subinterval

Let $[t_i, t_{i+1}[$ be the considered subinterval. In the spatial error term, only the term $\mathbf{H}(\mathbf{x}, t)$ has a time dependency as the metric field is fixed on the considered subinterval. Its integral over time $\int_{t_{i-1}}^{t_i} \mathbf{H}(\mathbf{x}, t)\mathrm{d}t$ has thus to be estimated. Here, we have two choices. Either, we can overestimate this integral using the L^∞ norm in time :

$$\int_{t_{i-1}}^{t_i} \mathbf{H}(\mathbf{x}, t)\mathrm{d}t \;\leq\; (t_i - t_{i-1}) \max_{t\in[t_i,t_{i+1}[} \mathbf{H}(\mathbf{x}, t) \;=\; \mathbf{H}^i_{L^\infty}(\mathbf{x}, t)\,.$$

This choice has a better detection of the high temporal variations of the unknowns. Or, the integral can be estimated using the Trapezoidal rule :

$$\int_{t_{i-1}}^{t_i} \mathbf{H}(\mathbf{x}, t)\mathrm{d}t \;\approx\; \sum_{k=1}^{n_{step}} (t_{k+1} - t_k)\, \frac{\mathbf{H}(\mathbf{x}, t_{k+1}) + \mathbf{H}(\mathbf{x}, t_k)}{2} \;=\; \mathbf{H}^i_{L^1}(\mathbf{x}, t)\,.$$

This choice is mandatory when considering the goal-oriented approach. In the following, we denote by $\mathbf{H}^i(\mathbf{x}, t)$ one of these two formulations.

The error in space for the considered subinterval becomes

$$\mathcal{E}^i_{\text{space}}(\mathcal{M}^i) = \int_\Omega \left(\text{trace}\left((\mathcal{M}^i(\mathbf{x}))^{-\frac{1}{2}}\mathbf{H}^i(\mathbf{x})(\mathcal{M}^i(\mathbf{x}))^{-\frac{1}{2}}\right)\right)^p \,\mathrm{d}\mathbf{x}\,,$$

and the spatial optimization problem on the subinterval reads

$$\mathcal{M}^i_{opt} \;=\; \underset{\mathcal{M}^i}{\text{Arg min}}\; \mathcal{E}^i_{\text{space}}(\mathcal{M}^i) \quad \text{such that} \quad \mathcal{C}(\mathcal{M}^i) = \int_\Omega \sqrt{\det \mathcal{M}^i(\mathbf{x})}\,\mathrm{d}\mathbf{x} = \mathcal{N}^i\,.$$

As previously, the optimal continuous mesh of this problem is

$$\mathcal{M}^i_{\text{opt}}(\mathbf{x}) = \left(\mathcal{N}^i\right)^{\frac{2}{d}} \left(\int_\Omega \left(\det \mathbf{H}^i(\mathbf{x})\right)^{\frac{p}{2p+d}} \,\mathrm{d}\mathbf{x}\right)^{-\frac{2}{d}} \left(\det \mathbf{H}^i(\mathbf{x})\right)^{-\frac{1}{2p+d}} \,\mathbf{H}^i(\mathbf{x})\,, \tag{8.21}$$

and the corresponding optimal error writes

$$\mathcal{E}^i_{\text{space}}(\mathcal{M}^i_{\text{opt}}) = d^p K^p_x \left(\mathcal{N}^i\right)^{-\frac{2p}{d}} \left(\int_\Omega \left(\det \mathbf{H}^i(\mathbf{x})\right)^{\frac{p}{2p+d}} \,\mathrm{d}\mathbf{x}\right)^{\frac{2p+d}{d}}\,.$$

5.4 Temporal optimization over the time subintervals

Now, the goal is to performed a global temporal minimization which will provide the optimal continuous mesh for each subinterval (in fact, the optimal spatial complexity for each subinterval as it is the only term which differs) and the optimal temporal mesh, i.e., the optimal time steps. We focus on minimizing the space-time error over the simulation time interval $[0, T]$:

$$\mathcal{E}^p_{\mathrm{st}}((\mathcal{M}^i)_i, \tau) = \sum_{i=1}^{n_{adap}} \left[\int_{t_{i-1}}^{t_i} K_t^p\, \tau(t)^{\alpha p} \left(\int_\Omega \left| W^*(\mathbf{x}, t) \frac{\partial^{\alpha+1} W(\mathbf{x}, t)}{\partial t^{\alpha+1}} \right|^p \mathrm{d}\mathbf{x} \right) + d^p K_x^p \left(\mathcal{N}^i \right)^{-\frac{2p}{d}} \left(\int_\Omega \left(\det \mathbf{H}^i(\mathbf{x}) \right)^{\frac{p}{2p+d}} \mathrm{d}\mathbf{x} \right.\right.$$

under the space-time complexity constraint

$$\mathcal{C}_{\mathrm{st}}((\mathcal{M}^i)_i, \tau) \;=\; \sum_{i=1}^{n_{adap}} \mathcal{N}^i \int_{t_{i-1}}^{t_i} (\tau(t))^{-1}\, \mathrm{d}t \;=\; \mathcal{N}_{\mathrm{st}}\,.$$

As previously, the optimization problem is reformulated in terms of $(\mathcal{N}^i)_i$ and τ by introducing two functions :

$$\begin{aligned}
\overline{\mathcal{G}}(\mathcal{N}^i, \tau) &= K_t^p\, \tau(t)^{\alpha p} \int_\Omega \left| W^*(\mathbf{x}, t) \cdot \frac{\partial^{\alpha+1} W(\mathbf{x}, t)}{\partial t^{\alpha+1}} \right|^p \mathrm{d}\mathbf{x}\,, \\
\mathfrak{F}(\mathcal{N}^i) &= d^p K_x^p \left(\mathcal{N}^i \right)^{-\frac{2p}{d}} \left(\int_\Omega \left(\det \mathbf{H}^i(\mathbf{x}) \right)^{\frac{p}{2p+d}} \mathrm{d}\mathbf{x} \right)^{\frac{2p+d}{d}}\,,
\end{aligned}$$

leading to

$$\left\{ \begin{array}{l}
\text{Find } ((\mathcal{N}^i_{\mathrm{opt}})_i, \tau_{\mathrm{opt}}) = \underset{(\mathcal{N})_i, \tau}{\mathrm{Arg\ min}} \displaystyle\sum_{i=1}^{n_{adap}} \left(\int_{t_{i-1}}^{t_i} \overline{\mathcal{G}}(\mathcal{N}^i, \tau)\, \mathrm{d}t + \mathfrak{F}(\mathcal{N}^i) \right), \\
\text{such that } \mathcal{C}_{\mathrm{st}}((\mathcal{N}^i_{\mathrm{opt}})_i, \tau) = \displaystyle\sum_{i=1}^{n_{adap}} \mathcal{N}^i \int_{t_{i-1}}^{t_i} \tau(t)^{-1}\, \mathrm{d}t = \mathcal{N}_{\mathrm{st}}.
\end{array} \right.$$

We consider the change of variables to avoid nonlinear constraints, namely $(\mathcal{N}^i,\, \zeta^i(t)) = (\mathcal{N}^i,\, \mathcal{N}^i\, \tau(t)^{-1})$, and we get

$$\mathcal{G}(\mathcal{N}^i, \zeta^i) = K_t^p\, \zeta^i(t)^{-\alpha p} \left(\mathcal{N}^i \right)^{\alpha p} \int_\Omega \left| W^*(\mathbf{x}, t) \cdot \frac{\partial^{\alpha+1} W(\mathbf{x}, t)}{\partial t^{\alpha+1}} \right|^p \mathrm{d}\mathbf{x}\,,$$

and the constraint becomes $\displaystyle\sum_{i=0}^{n_{adap}} \int_{t_{i-1}}^{t_i} \zeta^i(t)\, \mathrm{d}t = \mathcal{N}_{\mathrm{st}}$. Now, by setting

$$\begin{aligned}
\mathcal{U}(t) &= \alpha p\, K_t^p \int_\Omega \left| W^*(\mathbf{x}, t) \cdot \frac{\partial^{\alpha+1} W(\mathbf{x}, t)}{\partial t^{\alpha+1}} \right|^p \mathrm{d}\mathbf{x}\,, \\
\mathcal{K}^i \ &= 2p\, d^{p-1} K_x^p \left(\int_\Omega \left(\det \mathbf{H}^i(\mathbf{x}) \right)^{\frac{p}{2p+d}} \mathrm{d}\mathbf{x} \right)^{\frac{2p+d}{d}}\,,
\end{aligned}$$

the derivatives of functions $\mathfrak{F}$ and $\overline{\mathcal{G}}$ are given by

$$\begin{aligned}
\frac{\partial \mathfrak{F}}{\partial \mathcal{N}^i}\,\delta\mathcal{N}^i &= -\left(\mathcal{N}^i\right)^{-\frac{2p+d}{d}}\mathcal{K}^i\,\delta\mathcal{N}^i,\\
\frac{\partial \mathcal{G}}{\partial \mathcal{N}^i}\,\delta\mathcal{N}^i &= \zeta^i(t)^{-\alpha p}\left(\mathcal{N}^i\right)^{\alpha p-1}\mathcal{U}(t)\,\delta\mathcal{N}^i,\\
\frac{\partial \mathcal{G}}{\partial \zeta^i}\,\delta\zeta^i &= -\zeta^i(t)^{-\alpha p-1}\left(\mathcal{N}^i\right)^{\alpha p}\mathcal{U}(t)\,\delta\zeta^i.
\end{aligned}$$

The optimality condition writes

$$\left\{\begin{array}{l}
\displaystyle\sum_{i=1}^{n_{adap}}\left(\int_{t_{i-1}}^{t_i}\frac{\partial \mathcal{G}}{\partial \mathcal{N}^i}\,\mathrm{d}t+\frac{\partial \mathfrak{F}}{\partial \mathcal{N}^i}\right)\delta\mathcal{N}^i=0,\qquad \forall\delta\mathcal{N}^i\,,\\
\displaystyle\sum_{i=1}^{n_{adap}}\int_{t_{i-1}}^{t_i}\frac{\partial \mathcal{G}}{\partial \zeta^i}\delta\zeta^i\,\mathrm{d}t=0,\qquad \forall\delta\zeta^i\ \text{ such that }\ \sum_{i=1}^{n_{adap}}\int_{t_{i-1}}^{t_i}\delta\zeta^i\,\mathrm{d}t=0\,.
\end{array}\right.$$

From the above relations we deduce

$$\left\{\begin{array}{l}
\displaystyle\int_{t_{i-1}}^{t_i}\left(\frac{\partial \mathcal{G}}{\partial \mathcal{N}^i}\,\mathrm{d}t+\frac{\partial \mathfrak{F}}{\partial \mathcal{N}^i}\right)=0\,,\\
\displaystyle\frac{\partial \mathcal{G}}{\partial \zeta^i}=-C\,.
\end{array}\right.$$

where $C>0$ does not depend on time. The second equation gives

$$\mathcal{N}^i=\left(\frac{C}{\mathcal{U}(t)}\right)^{\frac{1}{\alpha p}}\zeta^i(t)^{\frac{\alpha p+1}{\alpha p}}\quad\text{or}\quad\zeta^i(t)=\left(\frac{\mathcal{U}(t)}{C}\right)^{\frac{1}{\alpha p+1}}\left(\mathcal{N}^i\right)^{\frac{\alpha p}{\alpha p+1}}.\tag{8.22}$$

Thanks to Equation (8.22), the first equation gives

$$\mathcal{N}^i=C^{-\frac{\alpha d}{\lambda}}\left(\mathcal{K}^i\right)^{\frac{d(\alpha p+1)}{\lambda p}}\left(\int_{t_{i-1}}^{t_i}\mathcal{U}(t)^{\frac{1}{\alpha p+1}}\,\mathrm{d}t\right)^{-\frac{d(\alpha p+1)}{\lambda p}},\ \text{ with }\lambda=2(\alpha p+1)+\alpha d\,.\tag{8.23}$$

Recalling that $\zeta^i(t)=\mathcal{N}^i\tau^{-1}(t)$ we have immediately

$$\tau^i(t)=\left(\frac{C}{\mathcal{U}(t)}\right)^{\frac{1}{\alpha p+1}}\left(\mathcal{N}^i\right)^{\frac{1}{\alpha p+1}}\ ,\tag{8.24}$$

then the constraint on the space-time complexity, using Equations (8.24) then (8.23) to replace $\tau(t)$ then $\mathcal{N}^i$, gives

$$\begin{aligned}
\mathcal{N}_{\text{st}}=\sum_{i=1}^{n_{adap}}\int_{t_{i-1}}^{t_i}\mathcal{N}^i(\tau(t))^{-1}\,\mathrm{d}t\ &=\ C^{-\frac{1}{\alpha p+1}}\sum_{i=1}^{n_{adap}}\left(\mathcal{N}^i\right)^{\frac{\alpha p}{\alpha p+1}}\int_{t_{i-1}}^{t_i}\mathcal{U}(t)^{\frac{1}{\alpha p+1}}\,\mathrm{d}t\\
C^{-\frac{\alpha d+2}{\lambda}}\sum_{i=1}^{n_{adap}}\left(\mathcal{K}^i\right)^{\frac{\alpha d}{\lambda}}\left(\int_{t_{i-1}}^{t_i}\mathcal{U}(t)^{\frac{1}{\alpha p+1}}\,\mathrm{d}t\right)^{\frac{2(\alpha p+1)}{\lambda}}&,
\end{aligned}$$

and we deduce

$$C \ = \mathcal{N}_{\text{st}}^{-\frac{\lambda}{\alpha d+2}} \left(\sum_{i=1}^{n_{adap}} \left(\mathcal{K}^i\right)^{\frac{\alpha d}{\lambda}} \left(\int_{t_{i-1}}^{t_i} \mathcal{U}(t)^{\frac{1}{\alpha p+1}}\, \mathrm{d}t \right)^{\frac{2(\alpha p+1)}{\lambda}} \right)^{\frac{\lambda}{\alpha d+2}} .$$

For readability, we introduce the global variable $\mathcal{S} = \sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{\alpha d}{\lambda}} \left(\int_{t_{i-1}}^{t_i} \mathcal{U}(t)^{\frac{1}{\alpha p+1}}\, \mathrm{d}t \right)^{\frac{2(\alpha p+1)}{\lambda}}$ such that $C = \mathcal{N}_{\text{st}}^{-\frac{\lambda}{\alpha d+2}} \mathcal{S}^{\frac{\lambda}{\alpha d+2}}$. Knowing the constant C, we can now express the solution of the optimization problem

$$\mathcal{N}_{\text{opt}}^i = \mathcal{N}_{\text{st}}^{\frac{\alpha d}{\alpha d+2}} \mathcal{S}^{-\frac{\alpha d}{\alpha d+2}} \left(\mathcal{K}^i\right)^{\frac{d(\alpha p+1)}{\lambda p}} \left(\int_{t_{i-1}}^{t_i} \mathcal{U}(t)^{\frac{1}{\alpha p+1}}\, \mathrm{d}t \right)^{-\frac{d(\alpha p+1)}{\lambda p}} ,$$

$$\tau_{\text{opt}}^i(t) = \mathcal{N}_{\text{st}}^{-\frac{2}{\alpha d+2}} \mathcal{S}^{\frac{2}{\alpha d+2}} \left(\mathcal{K}^i\right)^{\frac{d}{\lambda p}} \left(\mathcal{U}(t)\right)^{-\frac{1}{\alpha p+1}} \left(\int_{t_{i-1}}^{t_i} \mathcal{U}(t)^{\frac{1}{\alpha p+1}}\, \mathrm{d}t \right)^{-\frac{d}{\lambda p}} .$$

The optimal spatial continuous mesh for a given subinterval $[t_i, t_{i+1}[$ that minimizes the space-time error is simply obtained by using the above solution in Formula (8.21) :

$$\mathcal{M}_{\text{opt}}^i(\mathbf{x}) = \left(\mathcal{N}_{\text{opt}}^i\right)^{\frac{2}{d}} \left(\int_\Omega \left(\det \mathbf{H}^i(\mathbf{x})\right)^{\frac{p}{2p+d}} \, \mathrm{d}\mathbf{x} \right)^{-\frac{2}{d}} \left(\det \mathbf{H}^i(\mathbf{x})\right)^{-\frac{1}{2p+d}} \ \mathbf{H}^i(\mathbf{x}) \, .$$

This results shows that the space-time error model optimally distributes the number of vertices for each subinterval in order to minimize the space-time error.

Proposition 5.1 (Optimal space-time continuous mesh for the GTFP)
We consider the global space-time error model in L^p norm given by Equation (9.38). We define the time dependent function $\mathcal{U}$ and the variable $\mathcal{K}^i$ defined on each subinterval :

$$\begin{aligned} \mathcal{U}(t) \ &= \ \alpha p\, K_t^p \int_\Omega \left| W^*(\mathbf{x},t) \frac{\partial^{\alpha+1} W(\mathbf{x},t)}{\partial t^{\alpha+1}} \right|^p \mathrm{d}\mathbf{x} \, , \\ \mathcal{K}^i \ &= \ 2p\, d^{p-1} K_x^p \left(\int_\Omega \left(\det \mathbf{H}^i(\mathbf{x})\right)^{\frac{p}{2p+d}} \, \mathrm{d}\mathbf{x} \right)^{\frac{2p+d}{d}} \ , \end{aligned} \tag{8.25}$$

K_t and α depends on the chosen implicit time integration scheme, for instance for the second-order backward differencing formula we have $\alpha = 2$ and $K_t = \frac{1}{3}$. K_x depends on the spatial domain dimension d with $K_x = \frac{1}{8}$ in 2D and $K_x = \frac{1}{20}$ in 3D. We also introduce the following global variable :

$$\mathcal{S} \ = \ \sum_{i=1}^{n_{adap}} \left(\mathcal{K}^i\right)^{\frac{\alpha d}{\lambda}} \left(\int_{t_{i-1}}^{t_i} \mathcal{U}(t)^{\frac{1}{\alpha p+1}}\, \mathrm{d}t \right)^{\frac{2(\alpha p+1)}{\lambda}} ,$$

Then, for any subinterval $[t_i, t_{i+1}]$*, the optimal spatial mesh complexity* $\mathcal{N}^i_{\text{opt}}$ *and the optimal time step* $\tau^i_{\text{opt}}(t)$ *are given by :*

$$\mathcal{N}^i_{opt} = \mathcal{N}_{st}^{\frac{\alpha d}{\alpha d+2}} \mathcal{S}^{-\frac{\alpha d}{\alpha d+2}} \left(\mathcal{K}^i\right)^{\frac{d(\alpha p+1)}{\lambda p}} \left(\int_{t_{i-1}}^{t_i} \mathcal{U}(t)^{\frac{1}{\alpha p+1}}\,\mathrm{d}t\right)^{-\frac{d(\alpha p+1)}{\lambda p}}, \tag{8.26}$$

$$\tau^i_{opt}(t) = \mathcal{N}_{st}^{-\frac{2}{\alpha d+2}} \mathcal{S}^{\frac{2}{\alpha d+2}} \left(\mathcal{K}^i\right)^{\frac{d}{\lambda p}} \left(\mathcal{U}(t)\right)^{-\frac{1}{\alpha p+1}} \left(\int_{t_{i-1}}^{t_i} \mathcal{U}(t)^{\frac{1}{\alpha p+1}}\,\mathrm{d}t\right)^{-\frac{d}{\lambda p}}. \tag{8.27}$$

with $\lambda = 2(\alpha p+1) + \alpha d$*, and the optimal spatial continuous mesh is :*

$$\mathcal{M}^i_{\text{opt}}(\mathbf{x}) = \left(\mathcal{N}^i_{\text{opt}}\right)^{\frac{2}{d}} \left(\int_\Omega \left(\det \mathbf{H}^i(\mathbf{x})\right)^{\frac{p}{2p+d}}\,\mathrm{d}\mathbf{x}\right)^{-\frac{2}{d}} \left(\det \mathbf{H}^i(\mathbf{x})\right)^{-\frac{1}{2p+d}}\ \mathbf{H}^i(\mathbf{x})\,, \tag{8.28}$$

where $\mathbf{H}^i$ *is defined according to Section 5.3 with for the feature-based case* $\mathbf{H} = \left|H_{u_t} + H_u\right|$*, and for the goal-oriented case* $\mathbf{H}$ *is given by Relation (8.10) and* $p = 1$*.*

6 Global Space-Time Transient Fixed Point algorithm

The Global Transient Fixed Point (GTFP) algorithm was proposed for specifying automatically a *succession of* n_{adap} *meshes* over a decomposition in subintervals used for the transient process, see Figure 8.1. This algorithm needs to be extended to the space-time error analysis of the previous section where the temporal error and the adapted temporal mesh have to be also managed. This new algorithm, called Global Space-Time Transient Fixed Point (GSTTFP), is presented in Algorithm 8.2.

From a practical point of view, the flowchart presented in Figure 8.2 shows when the terms involved in Proposition 5.1 are computed. In a first phase (in purple), the successive time subintervals $[t_{i-1}, t_i[$ are visited for :

- (1a.) interpolating the final solution of the previous subinterval on the new subinterval mesh (if necessary),
- (1b.) computing the flow solution,
- (1c.) according to Equation (8.25), computing the spatial error $\mathcal{K}$
- (1d.) according to Equation (8.25), computing the temporal error $\mathcal{U}$.

In a second phase (in green), when all the time subintervals have been computed, the global variables are evaluated : (2.) the optimal spatial complexities $\mathcal{N}_{opt}$ (Eq. (8.26)), (3.) the optimal time steps τ_{opt} (Eq. (8.27)), and (4.) the optimal continuous meshes $\mathcal{M}_{opt}$ (Eq. (8.28)). Finally, (5.) a new discrete space-time mesh is generated.

Algorithm 8.2 – GST : Global Space-Time Transient Fixed Point for Unsteady Flows

```
Initial mesh, time step and solution $(\mathcal{H}^0, \tau^0, \mathcal{S}_0^0)$ and set targeted space-time complexity $\mathcal{N}_{\text{st}}$
# Fixed-point loop to converge the global space-time mesh adaptation problem
For $j = 1, n_{ptfx}$
    # Adaptive loop to advance the solution in time on time frame $[0, T]$
    1. For $i = 1, n_{adap}$    # Advance the solution in time in subinterval $[t_{i-1}, t_i[$
        (a) $\mathcal{S}_{0,i}^j$ = Interpolate conservatively next subinterval initial solution from $(\mathcal{H}_{i-1}^j, \mathcal{S}_{i-1}^j, \mathcal{H}_i^j)$ ;
        (b) $\mathcal{S}_i^j$ = Compute solution on subinterval from triple $(\mathcal{H}_i^j, \tau_i^j, \mathcal{S}_{0,i}^j)$ ;
        (c) $|\mathbf{H}|_i^j$ = Compute subinterval Hessian-metric from solution sample $(\mathcal{H}_i^j, \{\mathcal{S}_i^j(k)\}_{k=1,nk})$ ;
        (d) $(\mathcal{E}_{\text{time}})_i^j$ = Compute time error $(\mathcal{H}_i^j, \tau_i^j, \{\mathcal{S}_i^j(k)\}_{k=1,nk})$ ;
       EndFor
    2. $\{\mathcal{N}_{opt}^{i,j+1}\}_{i=1,n_{adap}}$ = Compute space complexity from $(\mathcal{S}_i^j, \{|\mathbf{H}|_i^j\}_{i=1,n_{adap}})$ ;
    3. $\{\tau_{opt}^{i,j+1}\}_{i=1,n_{adap}}$ = Compute time step from $(\mathcal{S}_i^j, \{|\mathbf{H}|_i^j\}_{i=1,n_{adap}})$ ;
    4. $\{\mathbf{M}_i^{j+1}\}_{i=1,n_{adap}}$ = Compute all metrics + gradation $(\{\mathcal{N}_{opt}^{i,j+1}\}_{i=1,n_{adap}}, \{|\mathbf{H}|_i^j\}_{i=1,n_{adap}})$ ;
    5. $\mathcal{H}_{\text{st}}^{j+1} = (\{\mathcal{H}_i^{j+1}, (t_k)_i^{j+1}\}_{i=1,n_{adap}})$ = Generate all subinterval adapted meshes and time steps $(\{\mathcal{H}_i^j,\ \mathbf{M}_i^j, \tau_{opt}^{i,j+1}\}_{i=1,n_{adap}})$ ;
EndFor
```

7 Numerical experiments

This paper focuses on unsteady flows that can be considered as quasi-steady where the unsteady turbulence evolves in a fixed local region. For instance, this is the case of the turbulence behind a cylinder. For this type of flow, it is interesting to consider the specific case of a single spatial adapted mesh for the entire simulation time frame. This type of approach is also very suitable for Large Eddy Simulation (LES) applications. This must be contrasted with true unsteady flows involving fast dynamics, such as blast wave applications, where considering many sub-intervals is mandatory to optimize the space-time mesh [13].

As regards the error estimate, we only consider the feature-based method with the local Mach number as sensor.

The Navier-Stokes system is discretized in space using a vertex-centered mixed finite

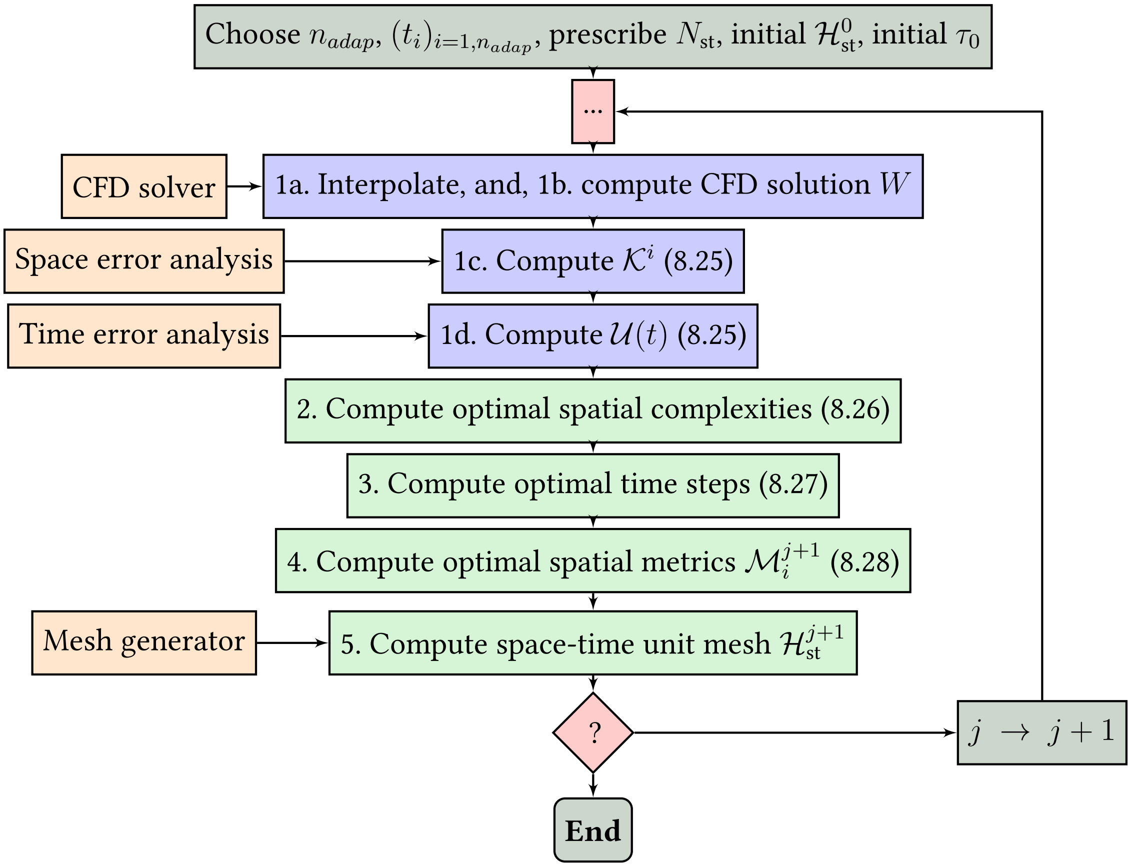


FIGURE 8.2 – Global Space-Time Transient Fixed Point (GSTTFP) flowchart. The “ ?” holds for testing whether the GSTTFP fixed point is converged or not.

volume/finite element numerical scheme applied to unstructured meshes composed of triangles/tetrahedra. The diffusive terms are discretized using P1 Galerkin finite-elements on the triangle/tetrahedra, whereas finite-volumes are used for the convective terms. The numerical approximation of the convective fluxes at the interface of neighboring finite volume cells is based on the HLLC approximate Riemann solver. To obtain second-order accuracy in space, the Monotone Upwind Scheme for Conservation Laws reconstruction method (MUSCL) is used. Time advancing is carried out through an implicit linearized method, based on a second-order accurate backward difference scheme, which means that in our theory $\alpha = 2$ and $K_t = \frac{1}{3}$ in Proposition 5.1.
As concerns mesh adaptation, we keep the ingredients of [10], except that the temporal error needs to be evaluated. It is a third derivative and, as any truncation error, most approximations are highly oscillating. We apply a strong moving average filtering. For all the simulations, the Spalart-Allmaras turbulence model is used [125]. More details can be found in [9, 39].

For all the presented simulations, the physical conditions are : $|\mathbf{u}| = 1\, m/s, \rho =$

$1\,kg/m^3$, $\mu = \frac{1}{Re}$ where Re is the Reynolds number. The Mach number is set to 0.3, so the pressure is $p \approx 7.9365\,Pa$. Our experience with vortex shedding flows is that using a unique adapted mesh for the several vortex shedding cycles lands to easier flow and error statistics.

7.1 2D flow past a cylinder at Reynolds 3900

The first test case is the 2D computation of a flow around a cylinder at Reynolds number 3900. A circular computational domain of radius 20 diameter of the cylinder is considered. The solution is initialized by running the solution for 200 seconds in physical time on an initial radial mesh composed of $12K$ vertices. Then, the simulation consists in running a period of 10 seconds which corresponds to two vortex shedding cycles. The space-time mesh adaptation is carried out for this 10 seconds time frame by using a single adapted spatial mesh, that is $n_{adap} = 1$. Four space-time complexity values were considered, namely $N_{\rm st}$ equal to 5M, 10M, 20M and 40M.

Figure 8.3 shows the obtained adapted spatial mesh and the associated final solution for a space-time complexity of 5M and 40M. We note that the mesh is highly refined in the boundary layer region and in the turbulent wake of the cylinder. The adapted mesh is anisotropic in the boundary layer while it is almost isotropic in the wake.

Figure 8.4 shows the computed time steps as functions of the physical time, i.e., the

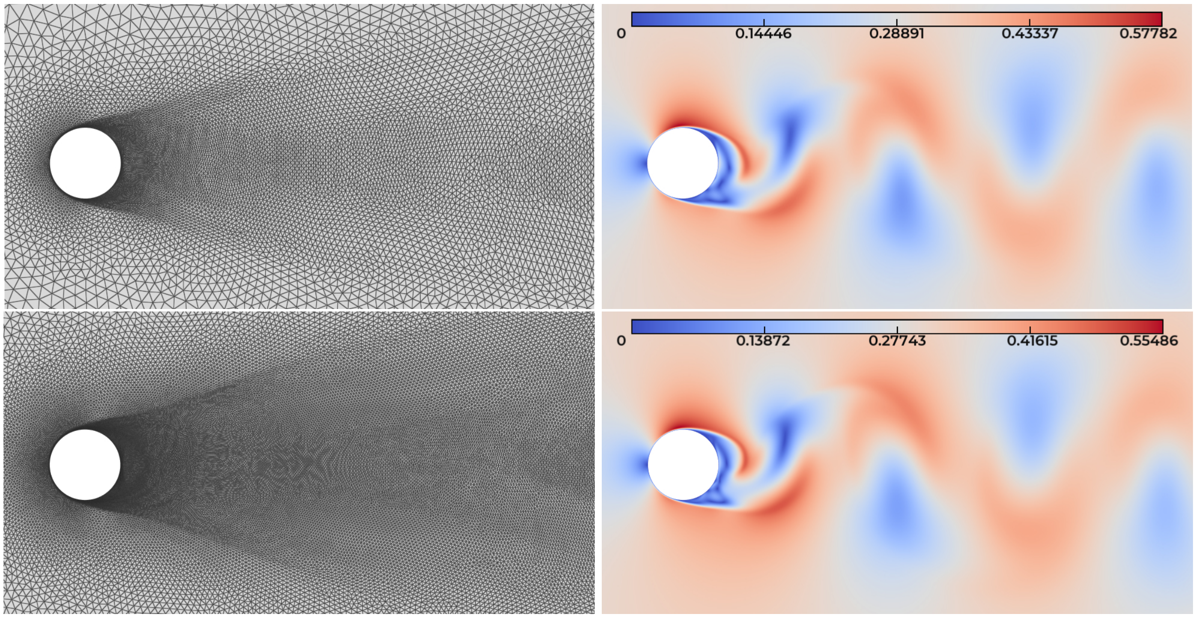


FIGURE 8.3 – 2D cylinder flow at Reynolds 3900. Adapted mesh (left) and the associated Mach number (right) for a space-time complexity of 5M (top) and 40M (bottom).

computed adapted temporal mesh, for the successive iterations of the fixed point, and this, for the four space-time complexities. We note that, after the second fixed point iteration, the curves are more or less identical. This points out the convergence of the GSTTFP algorithm. When we compare the computed time steps for the four complexities, we notice that the curves are very similar with a smaller amplitude for the largest space-time complexities corresponding to a higher time accuracy. It is interesting to note that we observe the quasi-periodicity of the flow in the temporal mesh - four periods are clearly visible which correspond to two vortex shedding cycles - despite the fact that no specific physical criteria have been introduced into the error estimation. This demonstrates that the error estimate is capture the physics of the flow.

By comparison with the previous version of the GTFP algorithm which does not adapt the time step, the extra computations are negligible. Conversely, the optimal choice of time step may induce an important saving in CPU time. As concerns the CPU time, each of the 15 iterations of the fixed point adaptative algorithm for the 40M space-time complexity takes six minutes in serial on a laptop Dell Precision of 2.3 GHz.

Figure 8.5 presents the evolution of the spatial $\mathcal{E}_{\text{space}}$ and the time $\mathcal{E}_{\text{time}}$ error functionals for the successive adaptation fixed point iterations. When the algorithm starts, we note a quick convergence of the process as only three to four fixed point iterations are sufficient to reach the optimal errors values. This result leads to two remarks.

First, whatever the space-time complexity, both errors converge toward similar values which is in agreement with our theory where the aim is to balance both errors. Indeed, if it is not the case, for instance if the temporal error is larger that the spatial one, this means that we could reduce slightly the number of vertices, increasing slightly the spatial error, and increase the number of time steps (decreasing the temporal error) with the result of making theses two errors closer. However, in the proposed method, the optimization is performed in a continuous context, producing an optimality condition, which we in turn discretize. The final difference between spatial and temporal errors is therefore due to the discretization (in space and time). The result obtained confirms this argument because we observe that when the space-time complexity increases from 5M to 40M, the final gap between the two errors decreases quickly.

Second, we can analyze the convergence order of the spatial and temporal errors between the space-time complexity of 5M and 40M. Between both complexities, the size of the space-time mesh, which is of dimension 3 (2D+t) is increased by a factor 8. For the errors, the reduction factor is $0.017/0.005 = 3.4$ from which we deduce a convergence order [2] of 1.76, close to second order.

---

[2] We have $\mathcal{E} \sim C\, N^{-\frac{\alpha}{d}}$ where α is the order of convergence, thus $\alpha = -d\frac{ln(\mathcal{E}_2/\mathcal{E}_1)}{ln(N_2/N_1)}$.

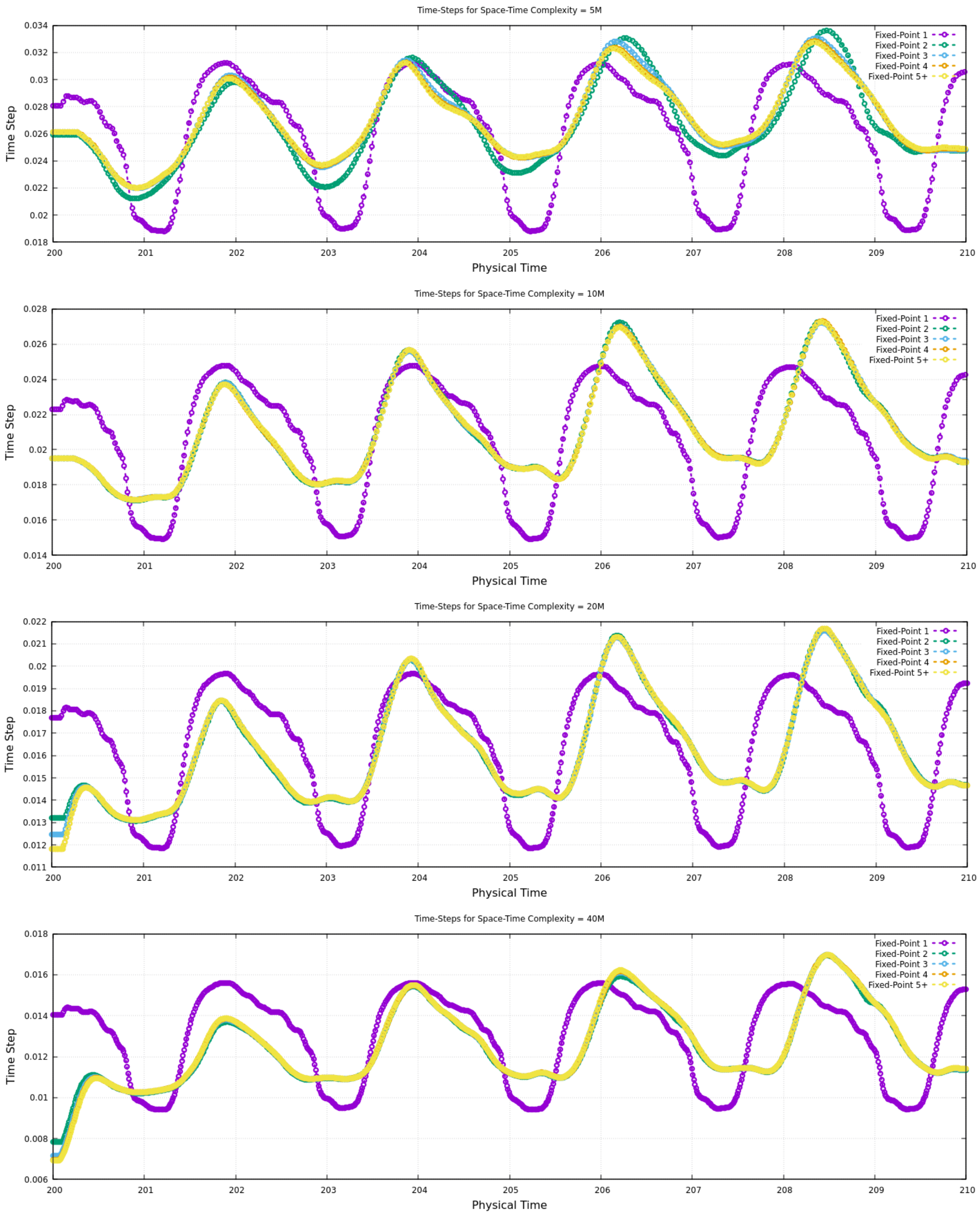


Figure 8.4 – 2D cylinder flow at Reynolds 3900. Computed time step as a function of the physical time for space-time complexities 5M, 10M, 20M, 40M (from top to bottom). The convergence of the computed temporal discretization for the successive iterations of the fixed point is depicted.

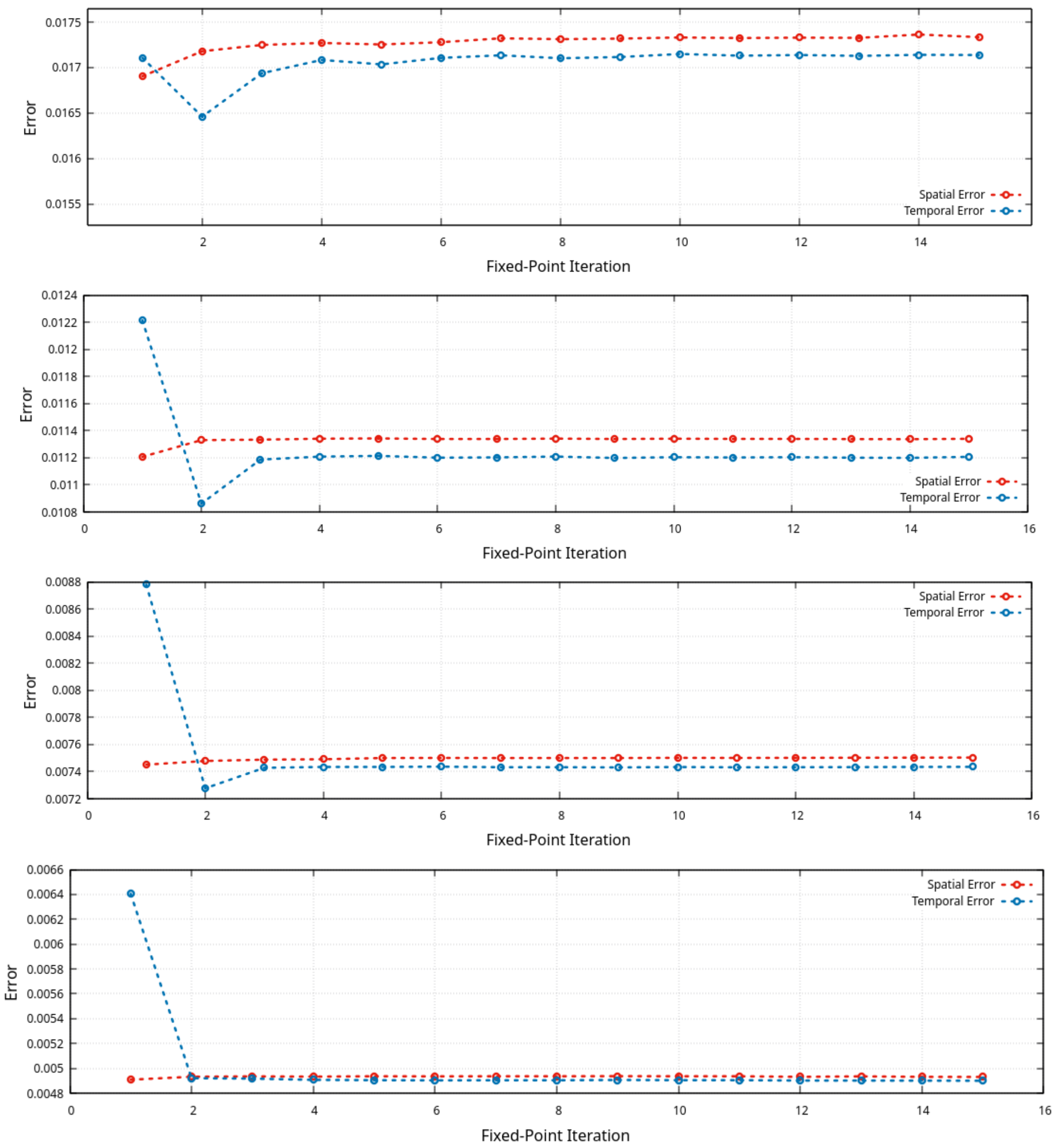


FIGURE 8.5 – 2D cylinder flow at Reynolds 3900. Continuous spatial and temporal errors for complexities 5M, 10M, 20M, 40M (from top to bottom) at each fixed point iteration. The corresponding total error levels at convergence are respectively 0.034, 0.022, 0.014, 0.01.

7.2 2D flow past a cylinder at Reynolds 3900 with multi-mesh adaptation

This section presents the computation of the previous test case when several different meshes on several time sub-intervals are used to adapt the space-time mesh, that is :

$$n_{adap} > 1.$$

The simulation parameters are then exactly the same as Section 7.1 (2D flow around a cylinder, Reynolds 3900, Spalart-Allmaras model, same initialisation, time interval is $[0, 10]$(seconds), 15 adaptation cycles) except n_{adap}, which is set to $n_{adap} = 2$ when the 5 M complexity is run, $n_{adap} = 4$ when the 10 M complexity is run, $n_{adap} = 8$ when the 20 M complexity is run, and $n_{adap} = 16$ when the 40 M complexity is run.

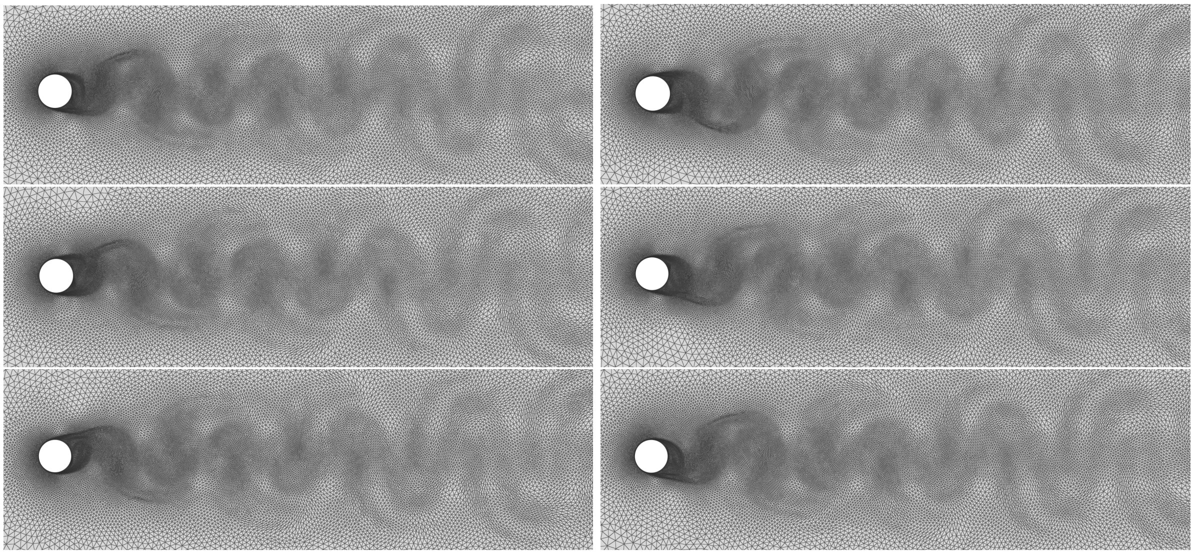


Figure 8.6 – 2D cylinder flow at Reynolds 3900 with multi-mesh adaptation. Final adapted meshes for a space-time complexity of 40M with 16 subintervals. View of the subinterval meshes 1, 4, 7, 10, 13, 16 (from left to right and from top to bottom) after the 15 fixed point iterations.

A subset of the 16 final adapted meshes is shown in Figure 8.6. As $n_{adap} = 16$ and the 10 seconds simulation corresponds to two vortex shedding cycles, 8 different adapted meshes are used to simulate one vortex shedding cycle. As a result, we clearly see the vortex shedding phenomena inside the adapted meshes even if, in this case, the mesh size is on average 30K vertices while it was 60K vertices with $n_{adap} = 1$. Moreover, we observe also in Figure 8.6 the progressive refinement of the wake through the simulation of the last fixed point iteration.

In contrast to the $n_{adap} = 1$ calculation, a temporal discretization of $1\,626$ time steps has computed which is more or less twice the number (825) of time steps of the $n_{adap} = 1$ case. If we analyze the computed temporal discretization in Figure 8.7, the four periods are also visible but the curve is less smooth showing that some specific details in the solution are more accurately captured. When we examine the time dependence of the optimum time step length and compare to the one-mesh calculation, the multi-mesh optimum time step length seems importantly perturbed by the 15 interfaces between subintervals.

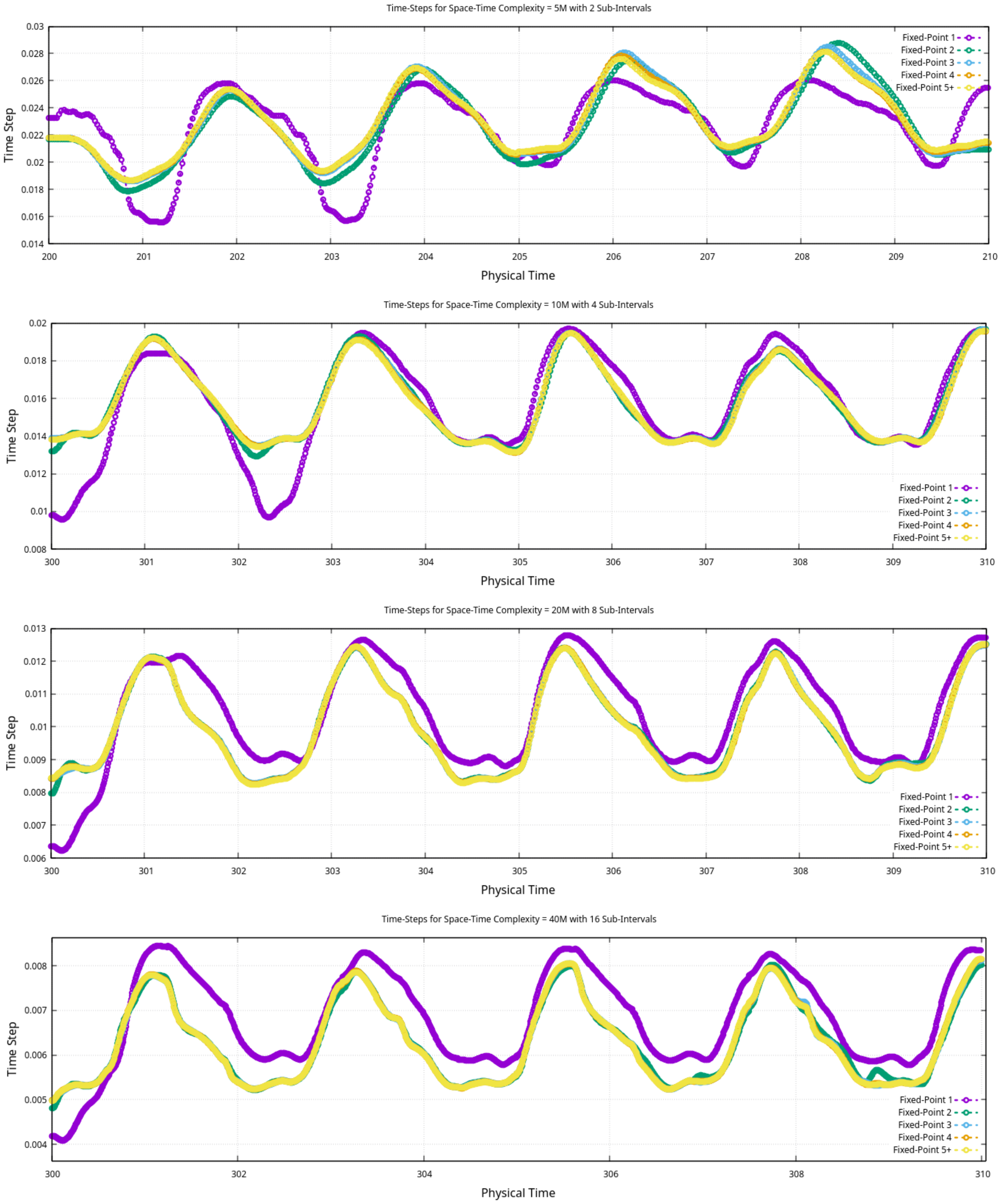


FIGURE 8.7 – 2D cylinder flow at Reynolds 3900 with multi-mesh adaptation. Computed time step as a function of the physical time for space-time complexity 40M. The convergence of the computed temporal discretization for the successive iterations of the fixed point is depicted.

Table 8.1 compares the numbers of both simulations.

Method	Specified space-time complexity	Optimal space complexity	Optimal number of vertices	Optimal number of time steps
One mesh	40M	48K	60K	825
Multi-mesh	40M	25K	30K	1626

Table 8.1 – 2D cylinder flow at Reynolds 3900. Statistics of the one-mesh ($n_{adap} = 1$) and multi-mesh ($n_{adap} = 16$) simulations.

Figure 8.8 shows a plot of the total error $\mathcal{E}_{\text{st}}$ for our computations described above, in respect of the number of sub-intervals 2, 4 ,8 and 16 for the respective complexity 5M, 10M, 20M and 40M. Table 8.2 compares the total error obtained when choosing a single mesh or several meshes for the experiment we have described. The total error is improvent by the multimesh option but the gain is small due to difficulties with the present method to master the time error.

Method	5M	10M	20M	40M
One mesh	0.0344	0.0225	0.0149	0.00982
Multi-mesh	0.0294	0.0192	0.0131	0.00938

Table 8.2 – 2D cylinder flow at Reynolds 3900. Total error for the one-mesh and multi-mesh simulations.

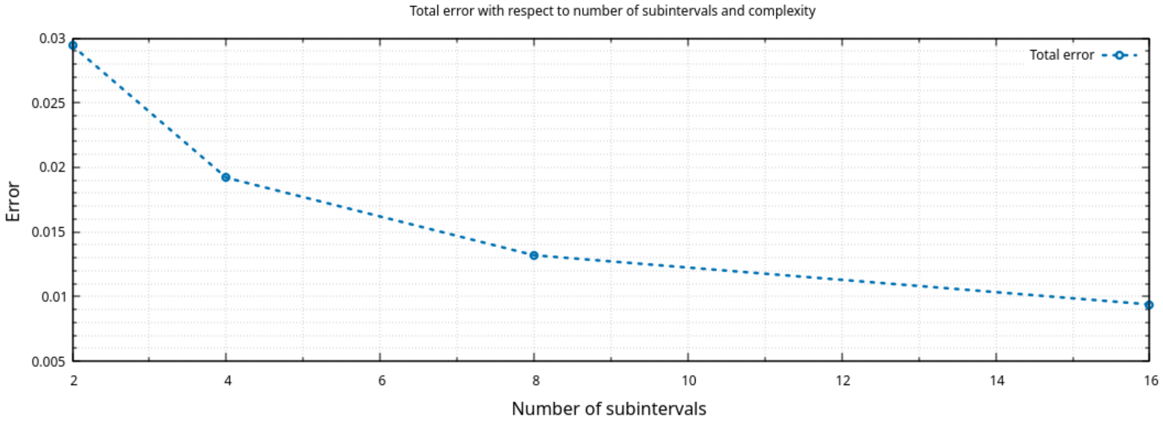


Figure 8.8 – 2D cylinder flow at Reynolds 3900 with multi-mesh. Continuous total errors for complexities 5M and 2 sub-intervals, 10M and 4 sub-intervals, 20M and 8 sub-intervals, 40M and 16 sub-intervals for the last fixed point iteration (of each computation).

7.3 2D flow past a cylinder at Reynolds 1M

The second test case is the 2D computation of a flow around a cylinder at Reynolds number 1M. The same circular computational domain as above is used. The solution is initialized by running the solution for 145 seconds in physical time on an initial radial mesh composed of $40K$ vertices. Then, the simulation consists in running a period of 9.3 seconds which corresponds to two vortex shedding cycles. The space-time mesh adaptation is carried out for this 9.3 seconds time frame by using a single adapted spatial mesh, that is $n_{adap} = 1$. Again, four space-time complexity values were considered, namely N_{st} equal to 12.5M, 25M, 50M and 100M. Similar outputs are presented for the analysis.

Figure 8.9 shows the obtained adapted spatial mesh and the associated final solution for a space-time complexity of 12.5M and 100M. We note that the mesh is highly refined in the very thin boundary layer region and in the turbulent wake of the cylinder. The adapted mesh is highly anisotropic in the boundary layer (with aspect ration $\simeq 10^2$) while it is almost isotropic in the wake. We note that the wake region is narrower near the cylinder compared to the previous case.

The computed adapted temporal discretization for the successive iterations of the fixed point for each complexity is depicted in Figure 8.11. As previously, two fixed point

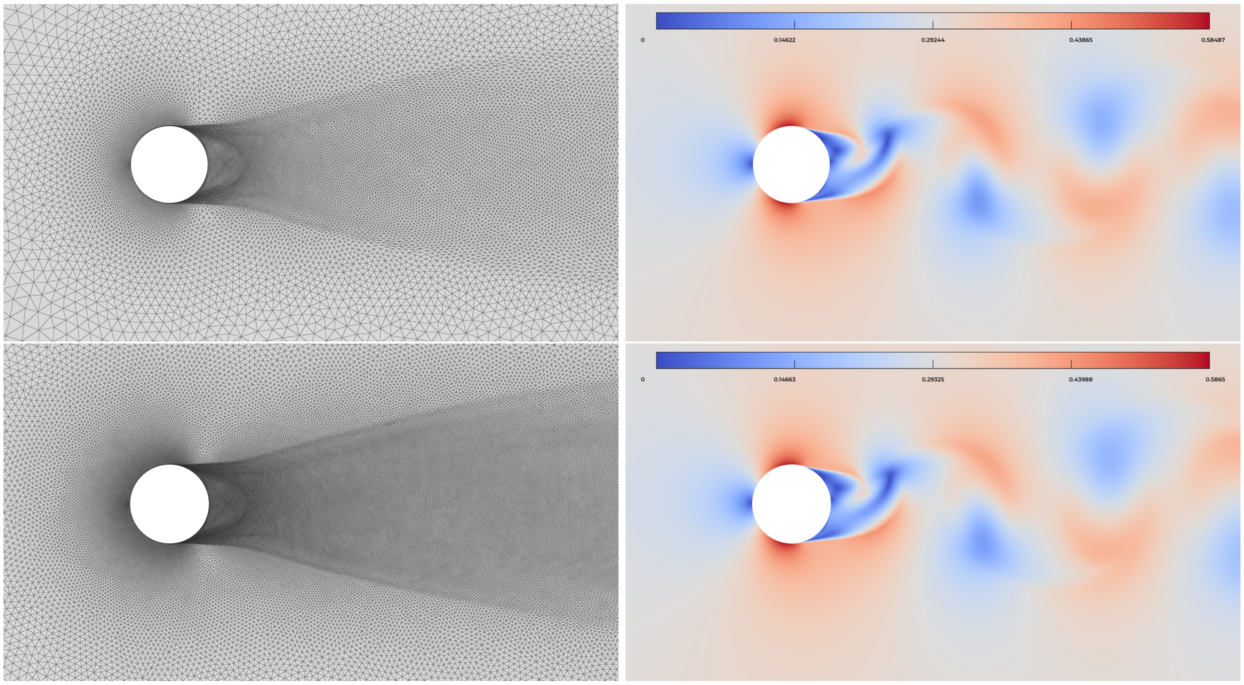


FIGURE 8.9 – 2D cylinder flow at Reynolds 1M. Adapted mesh (left) and the associated Mach number (right) for a space-time complexity of 12.5M (top) and 100M (bottom).

iterations are sufficient to converge the temporal adapted mesh. Whatever the complexity, the four temporal meshes are similar thus showing the same physics with five periods. The larger the complexity, the better the accuracy with smaller time steps. Note that for the finer mesh, a new detail in the solution is captured and it appears in the temporal mesh.

Figure 8.12 shows the evolution of the spatial $\mathcal{E}_{\text{space}}$ and the time $\mathcal{E}_{\text{time}}$ error functionals for the successive adaptation fixed point iterations. Again, the quick convergence of the process is clear as only two fixed point iterations are sufficient to reach the optimal errors values. As stated previously, in the proposed method, the optimization is performed in a continuous context, producing an optimality condition, which we in turn discretize. We may have a difference between spatial and temporal errors due to the discretization (in space and time). The difference between the temporal and spatial errors is of the order of 20%, which is acceptable. Note that the difference between the two errors in absolute value decreases when the space-time complexity increases : $\delta\mathcal{E} \sim 0.0012$ for $\mathcal{N}_{\text{st}} = 5M$ and $\delta\mathcal{E} \sim 0.0004$ for $\mathcal{N}_{\text{st}} = 40M$. This points out the mesh convergence of the GSTTFP process.

For this case, we obtain a convergence order of 1.81 of the space-time error between the space-time complexity of 12.5M and 100M.

Finally, Figure 8.10 plots the simulation CFL (min, max and average) and the minimal triangle's height of the mesh for the four complexities. We note that the process automatically sets an average CFL close 6500 while preserving the solution accuracy. At these high CFL values, implicit time integration schemes are a lot more efficient in CPU time than explicit schemes. Note that, when the mesh complexity increases, the smallest height of the mesh decreases (as we increase the mesh accuracy) and the computed time step decreases. But, the CFL stays almost the same whatever the complexity meaning that, depending on the physics, the GSTTFP algorithm automatically finds the optimal CFL to run the simulation. This demonstrates the powerfulness of the proposed method.

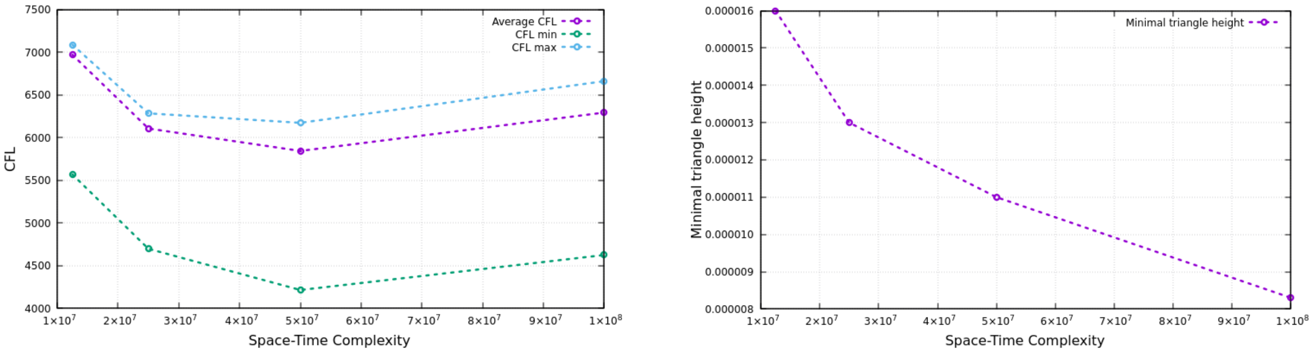


FIGURE 8.10 – 2D cylinder flow at Reynolds 1M. Evolution of CFL and the minima mesh height for the all the space-time complexities.

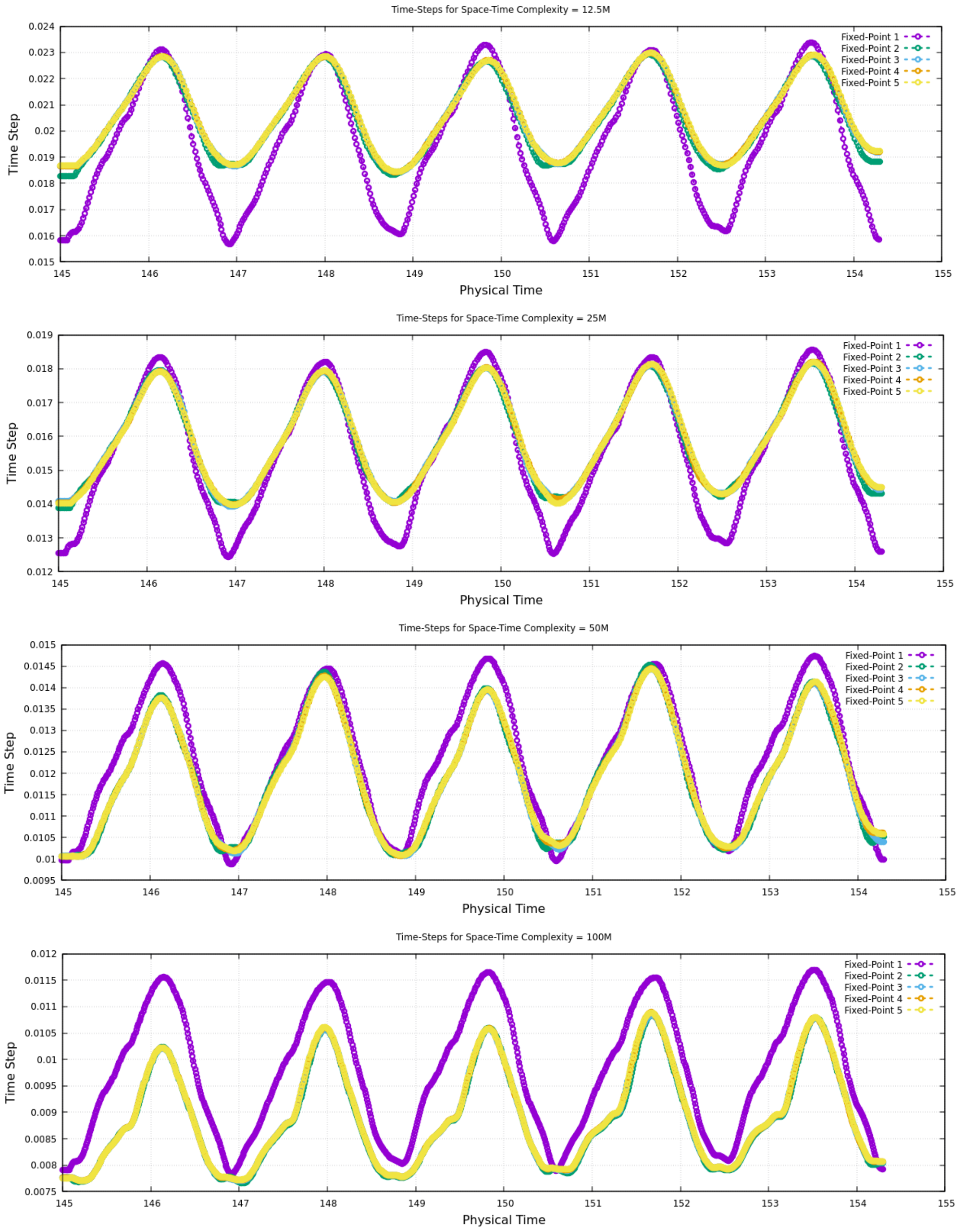


FIGURE 8.11 – 2D cylinder flow at Reynolds 1M. Computed time steps as a function of the physical time for space-time complexities 12.5M, 25M, 50M, 100M (from top to bottom). The convergence of the computed temporal discretization for the successive iterations of the fixed point is depicted.

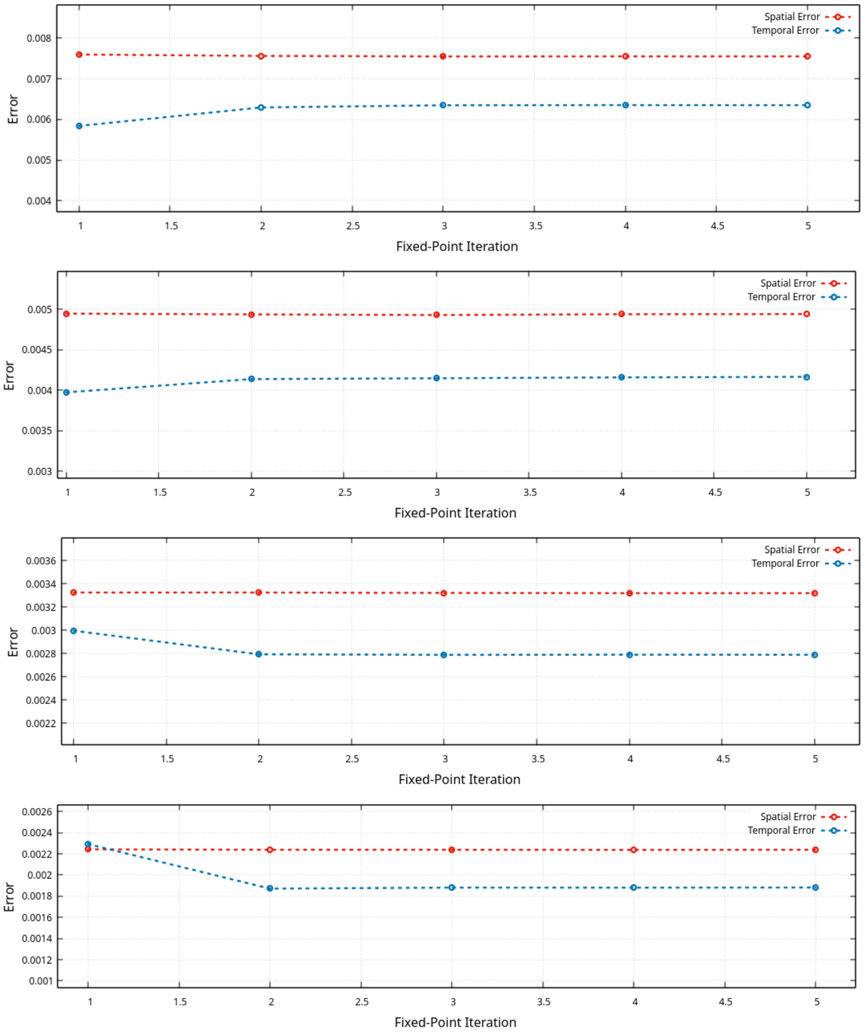


FIGURE 8.12 – 2D cylinder flow at Reynolds 1M. Continuous spatial and temporal errors for complexities 12.5M, 25M, 50M, 100M (from top to bottom) at each fixed point iteration. The corresponding total error levels at convergence are respectively 0.014, 0.009, 0.006, 0.004.

7.4 3D flow past a cylinder at Reynolds 3900

We consider again the first test case, but this time in 3D. The computational domain is cylindrical of radius 40 diameter of cylinder and a span of 3.14 diameters. The solution is initialized by running the solution for 140 seconds in physical time on an initial radial mesh composed of 840K vertices. The simulation consists in running a time frame of 20 seconds from the initialization. The space-time mesh adaptation is performed for this 20 seconds time frame by using a single adapted spatial mesh, that is $n_{adap} = 1$. The space-time complexity values is set to $N_{\text{st}} = 750$M.

The final spatial adapted mesh is rather coarse, it is composed of 1M vertices and the final temporal mesh has $1\,700$ time steps.

The spatial mesh accuracy is sufficient for the apparition of 3D features, as witnesses the examination of the Q factor, see Figure 8.13. Vortices are propagated in the wake without dissipation.

Figure 8.14 shows the computed time steps as functions of the physical time, i.e., the computed adapted temporal mesh, for the successive iterations of the fixed point. As in the 2D case, the temporal mesh is quickly converged. Two fixed point iterations are sufficient. The periodicity of the flow is again captured in the temporal mesh.

Figure 8.15 shows the evolution of the spatial $\mathcal{E}_{\text{space}}$ and the time $\mathcal{E}_{\text{time}}$ error functionals for the successive adaptation fixed point iterations. Similarly to the 2D case, we note a fast convergence of the process as only two fixed point iterations are sufficient to reach the optimal errors values. The temporal and spatial error are not perfectly balanced, there is a 20% difference. This is due to the fact that the optimization is performed in a continuous context, producing an optimality condition, which we in turn discretize. The difference between both errors will decrease when the space-time complexity increases.

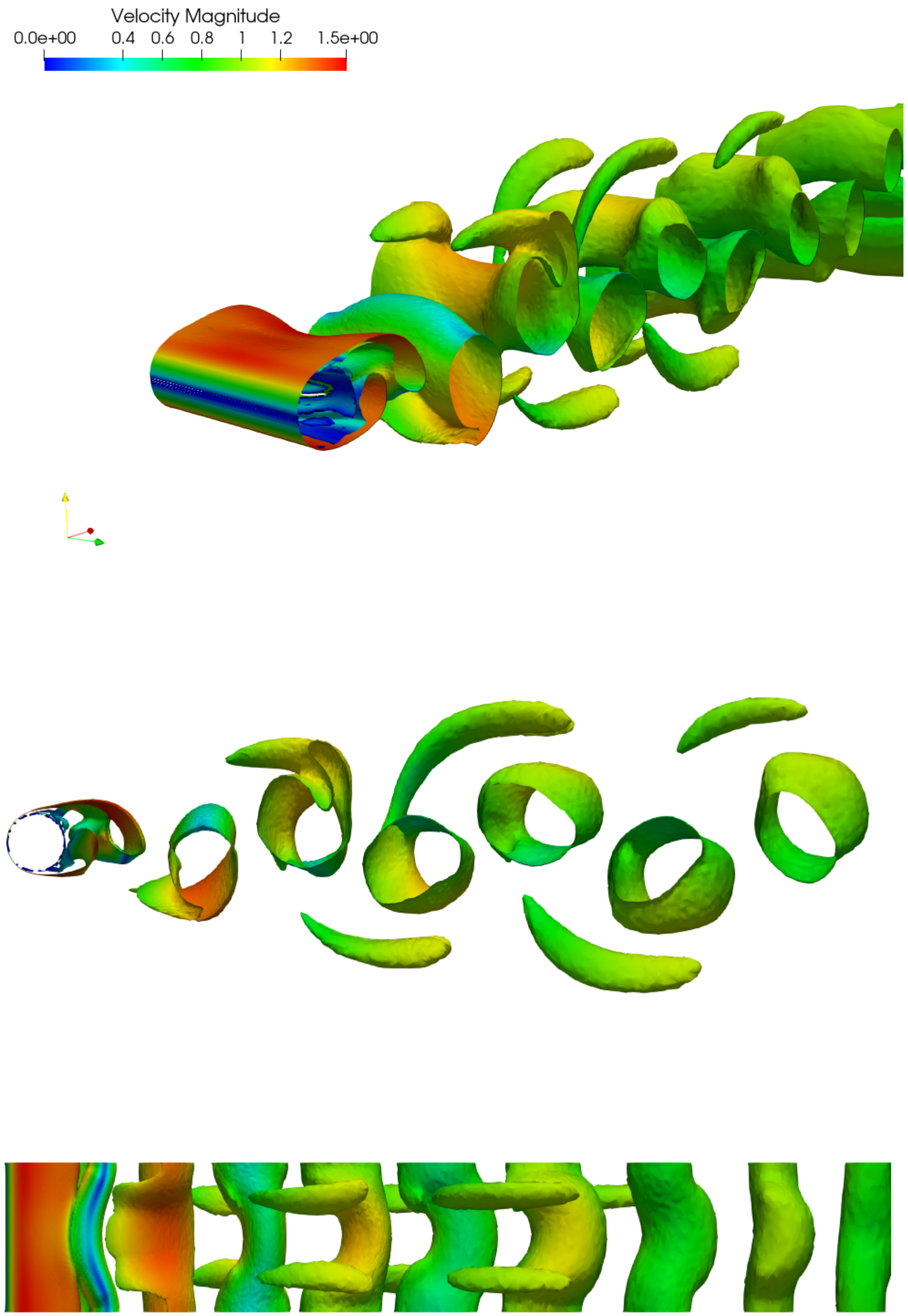


FIGURE 8.13 – 3D cylinder flow at Reynolds 3900. Different views of the Q-criterion isosurface colored with the velocity magnitude for the final solution.

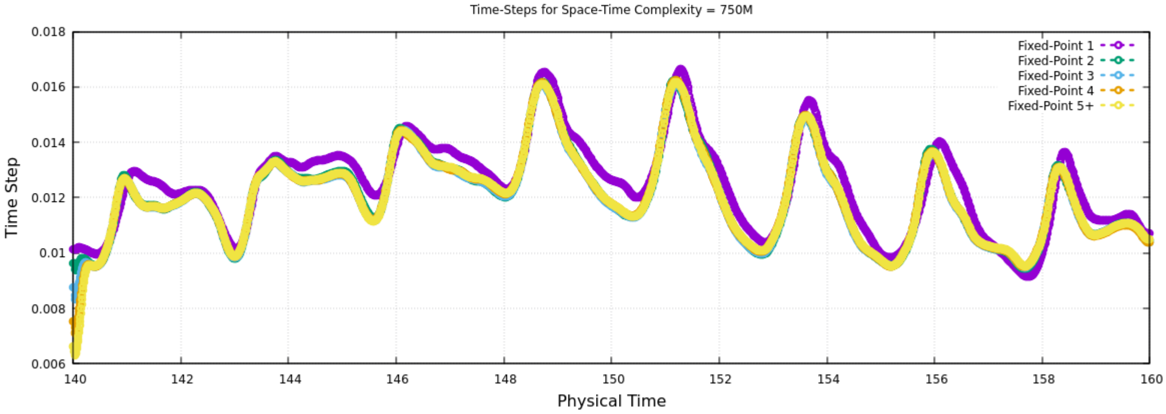


FIGURE 8.14 – 3D cylinder flow at Reynolds 3900. Computed time step as a function of the physical time for the space-time complexity 750M. The convergence of the computed temporal discretization for the successive iterations of the fixed point is depicted.

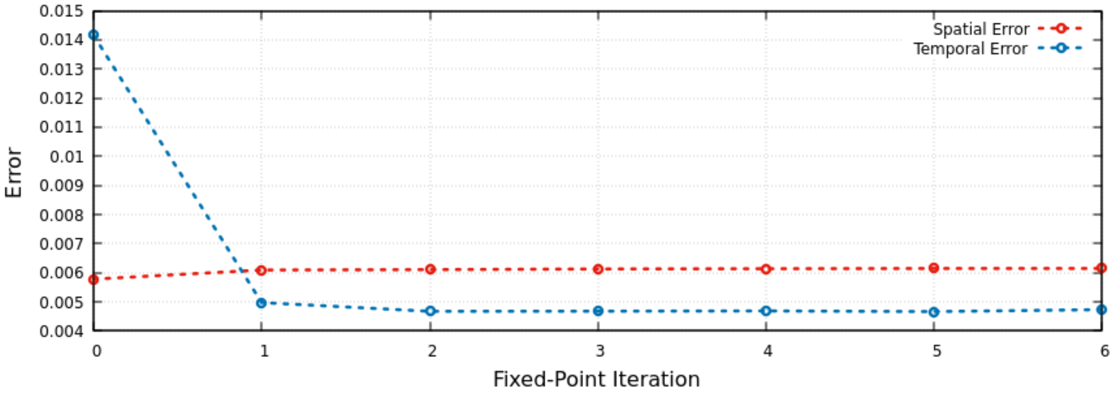


FIGURE 8.15 – 3D cylinder flow at Reynolds 3900. Continuous spatial and temporal errors for space-time complexity 750M at each fixed point iteration. The corresponding total error level at convergence is 0.011.

7.5 3D flow past a cylinder at Reynolds $1M$

Final, we consider the second test case in 3D. The computational domain is cylindrical of radius 40 diameter of cylinder and a span of 3.14 diameters. The solution is initialized by running the solution for 160 seconds in physical time on an initial radial mesh composed of 920K vertices. The simulation consists in running a time frame of 20 seconds from the initialization. The space-time mesh adaptation is performed for this 20 seconds time frame by using a single adapted spatial mesh, that is $n_{adap} = 1$. The space-time complexity values is set to $N_{\mathrm{st}} = 725$M.

The final spatial adapted mesh is rather coarse, it is composed of 1.2M vertices and the final temporal mesh has $1\,300$ time steps.

The spatial mesh accuracy is sufficient to capture the truly 3D flow. Figure 8.19 shows these 3D structures by plotting the Q-criterion. Vortices and other structures are propagated in the wake without dissipation.

Figure 8.16 shows the computed time steps as functions of the physical time, i.e., the computed adapted temporal mesh, for the successive iterations of the fixed point. We note that this 3D case is harder to converge, seven fixed point iteration are necessary to converge the temporal adapted mesh. The periodicity of the flow is again captured in the temporal mesh, we clearly see nine periods.

Figure 8.18 shows the evolution of the spatial $\mathcal{E}_{\mathrm{space}}$ and the time $\mathcal{E}_{\mathrm{time}}$ error functionals for the successive adaptation fixed point iterations. The convergence of the process is as only four fixed point iterations are sufficient to reach the optimal errors values. The temporal and spatial error are not perfectly balanced, there is a 30% difference again due to the discretization of the optimality conditions. But, the difference between both errors will decrease when the space-time complexity increases.

Figure 8.17 shows the evolution of the horizontal force applied to the cylinder and demonstrates the impact of the space-time adaptation. As a result, the drag coefficient, starting at a value of 0.58 after the first computation, takes a value of 0.52 after 10 space-time adaptation cycles.

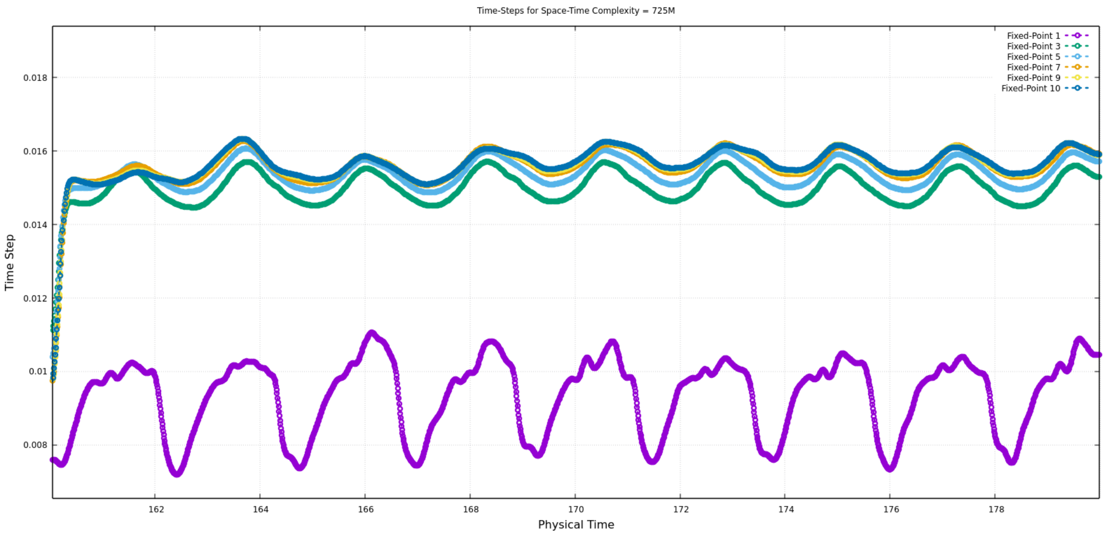


FIGURE 8.16 – 3D cylinder flow at Reynolds 1M. Computed time step as a function of the physical time for the space-time complexity 725M. The convergence of the computed temporal discretization for the successive iterations of the fixed point is depicted.

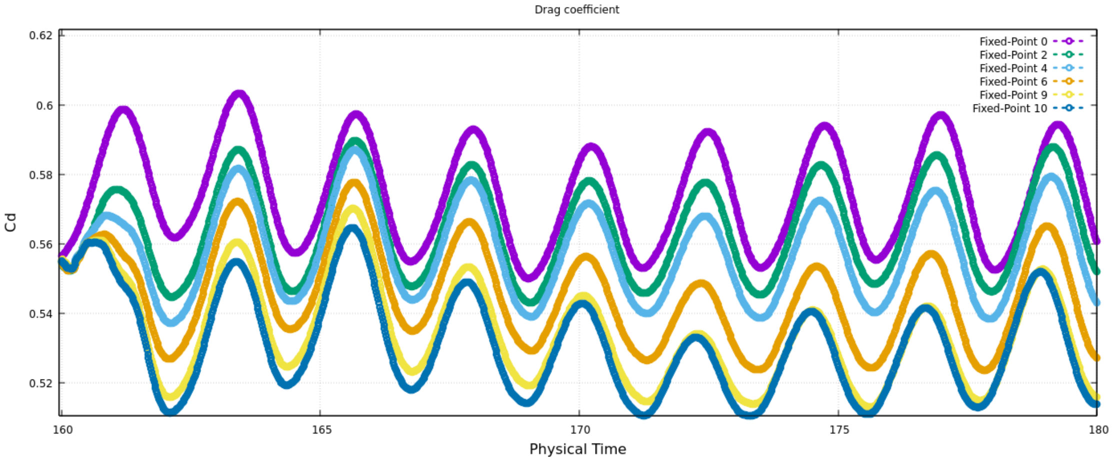


FIGURE 8.17 – 3D cylinder flow at Reynolds 1M. Drag coefficient evolution during the simulation time frame for the space-time complexity 725M. The convergence of the drag coefficient for the successive iterations of the fixed point is shown.

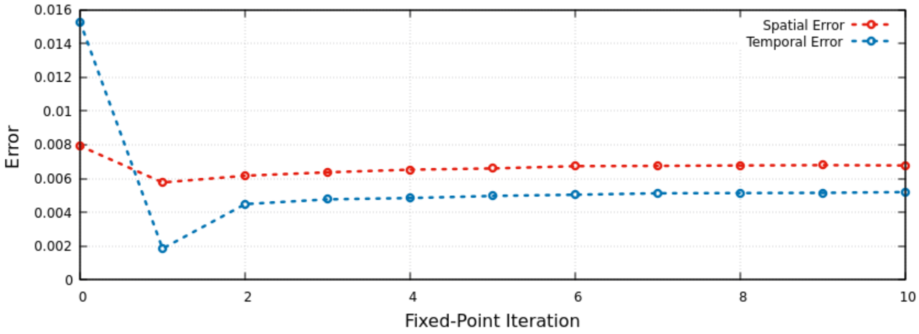


Figure 8.18 – 3D cylinder flow at Reynolds 1M. Continuous spatial and temporal errors for space-time complexity 725M at each fixed point iteration. The corresponding total error level at convergence is 0.012.

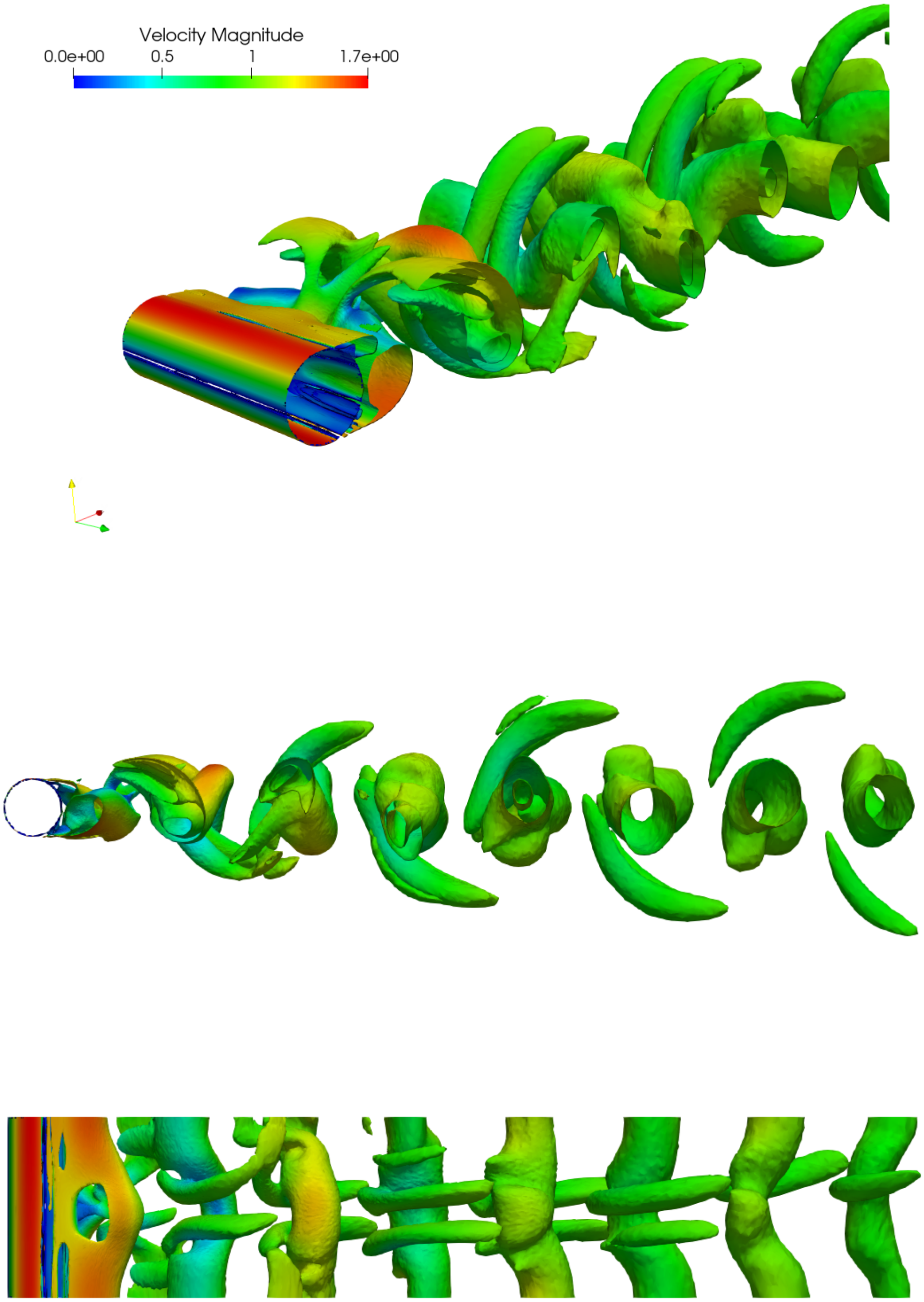


Figure 8.19 – 3D cylinder flow at Reynolds 1M. Different views of the Q-criterion isosurface colored with the velocity magnitude for the final solution.

8 Concluding remarks

During the computation of an unsteady RANS flow with an implicit time advancing, the choice of the time step is delicate and much influences both efficiency and accuracy. Manual prescription of the time-step requires expertise and may lead to inaccurate or expensive simulations when unknown configurations are considered. An automatic prescription of the time steps is therefore of main interest in an industrial context.

In this work, we have presented a Global Space-Time Transient Fixed Point (GSTTFP) method which automatically provides the optimal space-time adapted mesh, i.e., it provides the optimal spatial adapted mesh and the optimal time steps, to minimize the considered space-time error model. Two error models have been considered : the feature-based error model based on a control or the space-time interpolation error in L^p norm of a given sensor and the goal-oriented error model based on the control of the space-time approximation error of a given scalar output functional. This work extends the previous Global Transient Fixed Point methods (GTFP) proposed in [13] for the feature-based approach and [23] for the goal-oriented one. It should be noted that the new GSTTFP method can be directly used in place of the existing GTFP approach resulting in a direct CPU improvement induced by the optimal choice of time step.

The central principle of all these methods is to take into account the different approximation components in the combination of an unique error functional with a unique complexity constraint functional. This determines mathematically the optimal weighting between the different errors thanks to the derivation of optimality conditions. The coupling between the solver and the adaptation is then a discretization of the continuous optimality conditions. That discretization is in turn solved by a fixed-point iteration.

The proposed method, in the feature error model context, has been validated on 2D and 3D simulations of turbulent flow past a cylinder at different regimes. First, we have pointed out the fast convergence of the GSTTFP algorithm as just a few fixed point iterations are required to obtain the optimal spatial and temporal adapted meshes. In all cases, the spatial and temporal errors have been almost balanced by the algorithm. The gap between both error components decreases when the space-time complexity increases. We have also observed that the temporal adapted mesh captures the dynamic of the flow, in particular, the periodicity in time of the flow is clearly visible in the temporal adapted mesh. As regards the accuracy of the method, a space-time convergence order of ~ 1.8 has been observed for the 2D simulations using $n_{adap} = 1$. A higher order of convergence can be obtained by increasing progressively n_{adap}, see the analysis in [13].

The perspectives of this work are numerous.

First, the case where n_{adap} higher than unity requires a detailed numerical study for other types of applications involving, for instance, a rapid dynamics, although some results have been presented in this paper. In that context, the variable n_{adap} changes during the mesh convergence study (i.e., when the space-time complexity increases) in order to optimize the space-time mesh and increase the order of convergence according to the theoretical study described in [13, 23].

Second, the proposed error model is presented in a general formulation applying to either feature-based criteria or goal-oriented criteria. The important goal-oriented case needs further study and development as it requires the solution of the backward in time adjoint problem. This will be experimented in a future work.

Lastly, the proposed approach for quasi-steady flow using only one adapted mesh for the entire simulation time frame is very suitable for Large Eddy Simulation (LES) applications. The extension of this work to LES criteria will be considered in the future.

CHAPITRE 9

A metric-based mesh adaptation for hybrid RANS/LES flow calculations

1 Introduction

Large Eddy Simulations and hybrid flow calculations are today still very computer consuming CFD activities. More importantly, the user has difficuties in deciding if the mesh used allows the expected accuracy. Upgrading mesh adaptation methods to a better treatment of LES/hybrid flows is then an important issue in the Norma research.

Large Eddy Simulation (LES) is a numerical tool for predicting turbulent flows. Unlike a statistical Navier-Stokes model which tends to neglect fluctuations by applying a statistical average, which in most cases damps all these fluctuations, LES can be interpreted as damping only a part of unsteady turbulent structures, typically structures with a scale smaller than a prescribed filter size. The smallest structures being the most difficult to solve, LES consumes much less computational resources than a Direct Numerical Simulation which computes all turbulence structures. LES modelling relies on two steps :

(i)- defining the neglected scales as those which are smaller than a filter width Δ_f and adding a model of the action of neglected scales on the non-neglected ones. This defines a continuous model parameterized by the filter width Δ_f,

(ii)- using, in order to approximate the continuous model built in (i), a mesh-based approximation with local mesh size Δ_g, typically :

$$\Delta_g = (\xi\eta\zeta)^{\frac{1}{3}} \tag{9.1}$$

(expressed directly in terms of the local mesh sizes (ξ, η, ζ) measured in three orthogonal directions).

While, by construction, Δ_f should be larger than Δ_g, in order to approximate accurately the non-neglected scales, the research of the lowest computational cost motivates the practitioner to set

$$\Delta_f = \Delta_g,$$

with the consequence that the smallest unfiltered scales are the smallest scales computed on the grid and are then very poorly approximated, whatever be the accuracy of the numerical scheme.

When using a second-order accurate approximation, an important disadvantage comes from the fact that many LES models of Smagorinsky type are similar to second-order accurate truncation terms, in such a way that approximation errors are of same order as the filter model. According to an analysis of Ghosal [52] and to the outputs of many numerical computations, see e.g. [74], using a second-order accurate approximation may result in errors larger than the effect of LES modelling. It remains that second-order accurate approximations are much used and very useful for computing LES flows in engineering. A good practise for increasing the confidence in second-order accurate LES computations is to compare (a) the LES-based computations with (b) their no-model counterpart in order two, see e.g. [**?**].

Let assume that an approximation with a truncation error of order α [1] is used. An important issue is the fact that the convergence at truncation order is subject to the condition of using a sufficiently refined mesh : the mesh should be in any point sufficiently fine for capturing the smallest local detail of the flow computed, in order to start second-order or higher order convergence.

- Already in steady CFD, the convergence at truncation order is difficult to attain. A very efficient tool for obtaining this convergence is the convergent mesh adaptive double loop as described in [8, 39]. The inner loop is an anisotropic metric-based fixed-point *adaptation* working with a fixed number of unknowns. The outer loop is an anisotropic metric-based *enrichment* increasing progressively the total number of unknowns and controlling the actual convergence to the continuous solution. Thanks to this double loop, steady second-order RANS calculations are reaching a higher level of accuracy and fiability.
- As concerns unsteady RANS, mesh convergence with a double mesh adaptive loop is more difficult to apply, but effective in many cases. See for examples [13], [116], [106].
- As concerns mesh convergence with LES, it is a much more difficult issue. The scenario consisting of a brute-force strategy which increases simply the number of nodes generally may not succeed for the following reason. The filter term can be considered as a second order error. However, refining the mesh diminishes the subgrid-scale (SGS) term and introduces the arising of smaller and smaller new unstable scales in the solution which therefore cannot be accurately approximated until the process simply solves the corresponding DNS flow. Therefore, in contrast to laminar and RANS modeling, mesh adaptation for LES and hybrid models cannot have as goal the faster/fastest convergence to a continuous field, except the exact solution of Navier-Stokes.

The designing of a mesh adaptation criterion for LES is an important and difficult issue, addressed by many publications, among which we have selected the following typical ones.

---

[1] In the sense of the usual order of convergence which can be observed when no singularity occurs, typically second-order convergence when second-order codes are used with extremely fine meshes.

In [19], the approach is a numerical one, related to truncature. The error estimator identifies the regions lacking in accuracy, improving their resolution by either decreasing the size of the element or increasing the polynomial degree which approximates locally the solution. A smoothness indicator guides the hp-decision, leading to p-enrichment for smooth regions and h-refinement for non-smooth regions.

The work in [62] compares three indicators. The first indicator is based on the unsteady residual. The second indicator is based on a local smoothness indicator. The third indicator is based on an estimate for small scale turbulent kinetic energy. Comparisons with DNS tend to show that the first indicator is the best.

Similarly, in [59] several indicators more or less related to discretization and modeling error are compared with a wall jet as main test case.

In [46], a field-inversion machine-learning (FIML) framework is introduced. It only requires unsteady primal solutions. Two error estimates are compared in this work, a time-averaged unsteady residual weighted by a time-averaged adjoint, and an augmented-system residual weighted by the augmented-system adjoint.

In [95] the approach relies on a Discontinuous Galerkin (DG) high-order approximation. It does not really propose a mesh adaptation, but defines the ideal DG-LES solution as the result of the application of two successive filtering operations. A first convolution filter is applied to the DNS data which filters out frequencies beyond the LES grid cut-off. Next, a L^2-projection of this filtered field is performed on the hp-discretization space.

The physical approach is better addressed in [25] : arguments are based on the ratio of subgrid to viscous dissipation or viscosity. They are meaningful only in the buffer layer of wall-bounded turbulence while LES should be applicable to free shear flows at any Reynolds number.

A mesh adaptive strategy needs a satisfactory measure of the actual modelling error induced by LES. We discuss now an interesting analysis of this error. We have observed that in most models the local filter size introduced in practical LES models is generally taken identical to the local mesh size. Therefore informations concerning the improvement of the filter size can be useful for the improvement of mesh size. This is why we discuss now the work of Germano and co-workers [51] which have proposed a method for improving the filter size. We use the notations of [131] where this method is also explained. If a filter $W \mapsto \overline{W}$ of size $\bar{\Delta}$ is applied to the continuous (incompressible) Navier-Stokes equations for the exact field $(\mathcal{U}, \mathcal{P})$:

$$\mathcal{N}(\mathcal{U}) = \frac{\partial \mathcal{U}_i}{\partial t} + \frac{\partial \mathcal{U}_i \mathcal{U}_j}{\partial x_j} + \frac{1}{\rho}\frac{\partial \mathcal{P}}{\partial x_j} - \nu \frac{\partial^2 \mathcal{U}_i}{\partial x_j \partial x_j} = 0, \tag{9.2}$$

then the coarse-grained representation $(\overline{\mathcal{U}}, \overline{\mathcal{P}})$ of the exact field satisfies :

$$\begin{aligned} &\mathcal{N}^{\text{exact}}_{\bar\Delta}(\overline{\mathcal{U}}) = \frac{\partial \overline{\mathcal{U}}_i}{\partial t} + \frac{\partial \overline{\mathcal{U}}_i \overline{\mathcal{U}}_j}{\partial x_j} + \frac{1}{\rho}\frac{\partial \overline{\mathcal{P}}}{\partial x_j} - \nu \frac{\partial^2 \overline{\mathcal{U}}_i}{\partial x_j \partial x_j} + \frac{\partial \tau^{\text{exact}}_{ij,\bar\Delta}}{\partial x_j} = 0, \\ &\text{with} \quad \tau^{\text{exact}}_{ij,\bar\Delta} = \overline{\mathcal{U}_i \mathcal{U}_j} - \overline{\mathcal{U}}_i \overline{\mathcal{U}}_j . \end{aligned} \tag{9.3}$$

The basic idea of Germano's analysis is to introduce a second filter, not used directly in the model, the *filter test* with a size $\widetilde{\Delta} > \bar\Delta$ slightly larger than the LES filter. If we apply successively filter $\bar{.}$ and test filter $\tilde{.}$ to $\mathcal{U}$, we have

$$\mathcal{N}^{\text{exact}}_{\tilde{\bar\Delta}}(\tilde{\overline{\mathcal{U}}}) = \frac{\partial \tilde{\overline{\mathcal{U}}}_i}{\partial t} + \frac{\partial \tilde{\overline{\mathcal{U}}}_i \tilde{\overline{\mathcal{U}}}_j}{\partial x_j} + \frac{1}{\rho}\frac{\partial \tilde{\overline{\mathcal{P}}}}{\partial x_j} - \nu \frac{\partial^2 \tilde{\overline{\mathcal{U}}}_i}{\partial x_j \partial x_j} + \frac{\partial \tau^{\text{exact}}_{ij,\tilde{\bar\Delta}}}{\partial x_j} = 0,$$

while applying $\tilde{.}$ to $\mathcal{N}^{\text{exact}}_{\bar\Delta}(\overline{\mathcal{U}})$ gives :

$$\widetilde{\mathcal{N}^{\text{exact}}_{\bar\Delta}(\overline{\mathcal{U}})} = \frac{\partial \tilde{\overline{\mathcal{U}}}_i}{\partial t} + \frac{\partial \widetilde{\overline{\mathcal{U}}_i \overline{\mathcal{U}}_j}}{\partial x_j} + \frac{1}{\rho}\frac{\partial \tilde{\overline{\mathcal{P}}}}{\partial x_j} - \nu \frac{\partial^2 \tilde{\overline{\mathcal{U}}}_i}{\partial x_j \partial x_j} + \frac{\partial \widetilde{\tau^{\text{exact}}_{ij,\bar\Delta}}}{\partial x_j} = 0,$$

then

$$\mathcal{N}^{\text{exact}}_{\tilde{\bar\Delta}}(\tilde{\overline{\mathcal{U}}}) - \widetilde{\mathcal{N}^{\text{exact}}_{\bar\Delta}(\overline{\mathcal{U}})} = \frac{\partial}{\partial x_j}\Big(\widetilde{-\tau^{\text{exact}}_{ij,\bar\Delta}} + \tau^{\text{exact}}_{ij,\tilde{\bar\Delta}} - \widetilde{\overline{\mathcal{U}}_i \overline{\mathcal{U}}_j} + \tilde{\overline{\mathcal{U}}}_i \tilde{\overline{\mathcal{U}}}_j \Big). \tag{9.4}$$

In [51] the Germano identity is shown :

$$\widetilde{-\tau^{\text{exact}}_{ij,\bar\Delta}} + \tau^{\text{exact}}_{ij,\tilde{\bar\Delta}} - \widetilde{\overline{\mathcal{U}}_i \overline{\mathcal{U}}_j} + \tilde{\overline{\mathcal{U}}}_i \tilde{\overline{\mathcal{U}}}_j = 0.$$

and the RHS of (9.4) is zero. The Germano identity is true for continuous solutions of the exact Navier-Stokes system but will not apply if the exact Navier-Stokes system is replaced by a continuous formulation of a LES model *with one filter* with $\bar u$ as solution :

$$\mathcal{N}^{\text{model}}_{\bar\Delta}(\overline{u}) = \frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} + \frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j} - \nu \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j} + \frac{\partial \tau^{\text{model}}_{ij,\bar\Delta}(\overline{u})}{\partial x_j} = 0. \tag{9.5}$$

To fix the ideas, for the Smagorinsky model (incompressible case), it writes :

$$\begin{aligned} \tau^{\text{model}}_{ij,\bar\Delta}(\overline{u}) &= -(C_s \bar\Delta)^2 |S| P_{ij}, \\ S_{ij} &= \frac{1}{2}\left(\frac{\partial \bar u_i}{\partial x_j} + \frac{\partial \bar u_j}{\partial x_i} \right), \\ P_{ij} &= 2 S_{ij} - \tfrac{2}{3} S_{kk} \delta_{ij}, \end{aligned} \tag{9.6}$$

C_s being the Smagorinsky constant. A second formulation is the formulation *with both filters* with $\tilde{\overline{v}}$ as solution :

$$\mathcal{N}^{\text{model}}_{\tilde{\bar\Delta}}(\tilde{\overline{v}}) = \frac{\partial \tilde{\overline{v}}_i}{\partial t} + \frac{\partial \tilde{\overline{v}}_i \tilde{\overline{v}}_j}{\partial x_j} + \frac{1}{\rho}\frac{\partial \tilde{\overline{p}}}{\partial x_j} - \nu \frac{\partial^2 \tilde{\overline{v}}_i}{\partial x_j \partial x_j} + \frac{\partial \tau^{\text{model}}_{ij,\tilde{\bar\Delta}}(\tilde{\overline{v}})}{\partial x_j} = 0. \tag{9.7}$$

which implies that :

$$\widetilde{\mathcal{N}^{\text{model}}_{\bar{\Delta}}(\overline{u})} - \mathcal{N}^{\text{model}}_{\tilde{\bar{\Delta}}}(\tilde{\overline{u}}) = \frac{\partial}{\partial x_j}\Big(\widetilde{\tau^{\text{model}}_{ij,\bar{\Delta}}(\overline{u})} \;-\; \tau^{\text{model}}_{ij,\tilde{\bar{\Delta}}}(\tilde{\overline{u}}) + \widetilde{\overline{u}_i\overline{u}_j} - \tilde{\overline{u}}_i\tilde{\overline{u}}_j\Big) \tag{9.8}$$

which in general is not zero.

The interest of 9.8 is that it does not give an equation for the *model error* $u - \overline{u}$, but an equation for a quantity $\overline{u} - \tilde{\overline{u}}$ which is quite close to the model error.

Although discarding the divergence $\frac{\partial}{\partial x_j}$, the Germano identity error

$$\mathcal{G}_{ij} \;=\; \widetilde{\tau^{\text{model}}_{ij,\bar{\Delta}}(\overline{u})} \;-\; \tau^{\text{model}}_{ij,\tilde{\bar{\Delta}}}(\tilde{\overline{u}}) + \widetilde{\overline{u}_i\overline{u}_j} - \tilde{\overline{u}}_i\tilde{\overline{u}}_j$$

is a (tensorial) measure of the error between the (non-discretized) Navier Stokes flow field and the (non-discretized) LES flow field. Therefore, the Dynamic Germano-Piomelli procedure, by minimizing in some sense the Germano identity error, allows to find either the *optimal coefficient* C_s, or the *optimal product* $C_s\bar{\Delta}$ (both used in (9.6)). In the Dynamic Germano-Piomelli practice, $C_s = 0.1$, $\bar{u}$ is not known and is replaced by its discrete analog $\bar{u}_h$ computed on a given mesh, the filter size $\bar{\Delta}$ is generally chosen as the local mesh size, the filter test is chosen as $2\bar{\Delta}$, and the method produces a new product $(C_s\bar{\Delta})_{\text{opt}}$, which is usually interpreted as giving a new value $(C_s)_{\text{opt}} = (C_s\bar{\Delta})_{\text{opt}}/\bar{\Delta}$.

Assuming that we do not want to change C_s, the same computation proposes a new value for $\bar{\Delta}$:

$$(\bar{\Delta})_{\text{opt}} = (C_s\bar{\Delta})_{\text{opt}}/C_s$$

which seems to give informations concerning the filter, and then concerning the mesh to use.

The dynamic Germano analysis has inspired Toosi and Larsson [130] in proposing a method for adapting the mesh to a LES formulation. Toosi and Larsson identify *the source of LES-modelling error* as the residual of the governing LES equation applied to the coarse-grained exact Navier-Stokes solution. For filter level $\bar{\Delta}$:

$$\mathcal{R}_{i,\bar{\Delta}} \equiv \mathcal{N}^{\text{model}}_{\bar{\Delta}}(\overline{\mathcal{U}}) = \frac{\partial\overline{\mathcal{U}}_i}{\partial t} + \frac{\partial\overline{\mathcal{U}}_i\overline{\mathcal{U}}_j}{\partial x_j} + \frac{1}{\rho}\frac{\partial\overline{\mathcal{P}}}{\partial x_j} - \nu\frac{\partial^2\overline{\mathcal{U}}_i}{\partial x_j\partial x_j} + \frac{\partial\tau^{\text{model}}_{ij,\bar{\Delta}}(\overline{\mathcal{U}})}{\partial x_j}$$

and, using (9.3), we get

$$\mathcal{R}_{i,\bar{\Delta}} = \frac{\partial}{\partial x_j}\Big[\tau^{\text{model}}_{ij,\bar{\Delta}}(\overline{\mathcal{U}}) - \tau^{\text{exact}}_{ij,\bar{\Delta}}\Big].$$

Similarly, for filter test $\tilde{\bar{\Delta}}$:

$$\mathcal{R}_{i,\tilde{\bar{\Delta}}} \equiv \mathcal{N}^{\text{model}}_{\tilde{\bar{\Delta}}}(\widetilde{\overline{\mathcal{U}}}) = \frac{\partial}{\partial x_j}\Big[\tau^{\text{model}}_{ij,\tilde{\bar{\Delta}}}(\widetilde{\overline{\mathcal{U}}}) - \tau^{\text{exact}}_{ij,\tilde{\bar{\Delta}}}\Big]. \tag{9.9}$$

The residual $\mathcal{R}_{i,\tilde{\bar{\Delta}}}$ is chosen as the *error source of the LES modelling* but is expressed in terms of the unknown exact solution $\mathcal{U}$. In order to use it in practice, we need to replace $\mathcal{U}$ by an approximate evaluation of it. The *fundamental approximation* done by Toosi and Larsson is replacing in $\mathcal{N}^{\text{model}}_{\tilde{\bar{\Delta}}}$ the coarse-grained exact Navier-Stokes solution $\widetilde{\overline{\mathcal{U}}}$ by the LES one $\tilde{\bar{u}}$:

$$\mathcal{R}_{i,\tilde{\bar{\Delta}}} \equiv \mathcal{N}^{\text{model}}_{\tilde{\bar{\Delta}}}(\widetilde{\overline{\mathcal{U}}}) \approx \mathcal{N}^{\text{model}}_{\tilde{\bar{\Delta}}}(\tilde{\bar{u}}) \tag{9.10}$$

thus :

$$\mathcal{R}_{i,\tilde{\bar{\Delta}}} \approx \frac{\partial \tilde{\bar{u}}_i}{\partial t} + \frac{\partial \tilde{\bar{u}}_i \tilde{\bar{u}}_j}{\partial x_j} + \frac{1}{\rho}\frac{\partial \tilde{\bar{p}}}{\partial x_j} - \nu \frac{\partial^2 \tilde{\bar{u}}_i}{\partial x_j \partial x_j} + \frac{\partial \tau^{\text{model}}_{ij,\tilde{\bar{\Delta}}}(\tilde{\bar{u}})}{\partial x_j}$$

which is transformed applying the test-filtering to (9.5) :

$$\frac{\partial \tilde{\bar{u}}_i}{\partial t} + \frac{\partial \widetilde{\bar{u}_i \bar{u}_j}}{\partial x_j} + \frac{1}{\rho}\frac{\partial \tilde{\bar{p}}}{\partial x_j} - \nu \frac{\partial^2 \widetilde{\bar{u}_i}}{\partial x_j \partial x_j} + \frac{\partial \widetilde{\tau^{\text{model}}_{ij,\bar{\Delta}}(\bar{u})}}{\partial x_j} = 0,$$

and then substracting the result from (9.10). This gives :

$$\mathcal{R}_{i,\tilde{\bar{\Delta}}} \approx \mathcal{F}_{i,\tilde{\bar{\Delta}}} = \frac{\partial}{\partial x}\Big[\tau^{\text{model}}_{ij,\tilde{\bar{\Delta}}}(\tilde{\bar{u}}) \; - \; \widetilde{\tau^{\text{model}}_{ij,\bar{\Delta}}}(\bar{u}) - \widetilde{\bar{u}_i \bar{u}_j} + \tilde{\bar{u}}_i \tilde{\bar{u}}_j\Big]. \tag{9.11}$$

In order to minimize the RHS of (9.11), the authors consider a directional test filter $\tilde{\cdot}^{\mathbf{n}_x}$ of size $\bar{\Delta}_{\mathbf{n}_x}$, in direction $\mathbf{n}_x$:

$$\tilde{\bar{\phi}}^{(\mathbf{n}_x)} \approx \left(I + \frac{\bar{\Delta}^2_{\mathbf{n}_x}}{4}\mathbf{n}_x^T \nabla \nabla^T \mathbf{n}_x\right) \bar{\phi}. \tag{9.12}$$

Applying the directional test filter to equation (9.5) gives the following evolution equation for the filtered instantaneous fields at the filter test level :

$$\frac{\partial \tilde{\bar{u}}_i^{(\mathbf{n}_x)}}{\partial t} + \frac{\partial \widetilde{\bar{u}_i \bar{u}_j}^{(\mathbf{n}_x)}}{\partial x_j} + \frac{1}{\rho}\frac{\partial \tilde{\bar{p}}^{(\mathbf{n}_x)}}{\partial x_j} - \nu \frac{\partial^2 \tilde{\bar{u}}_i^{(\mathbf{n}_x)}}{\partial x_j \partial x_j} + \frac{\partial \widetilde{\tau^{\text{model}}_{ij}(\bar{u})}^{(\mathbf{n}_x)}}{\partial x_j} = 0. \tag{9.13}$$

Following the above calculation in this directional context, the following source term analog to (9.11) is obtained :

$$\tilde{\bar{\mathcal{F}}}_i^{(\mathbf{n}_x)}(\mathbf{x}) = \frac{\partial}{\partial x_j}\left(\tau^{\text{model}}_{ij}(\tilde{\bar{u}}^{(\mathbf{n}_x)}) - \widetilde{\tau^{\text{model}}_{ij}(\bar{u})}^{(\mathbf{n}_x)} - \widetilde{\bar{u}_i \bar{u}_j}^{(\mathbf{n}_x)} + \tilde{\bar{u}}_i^{(\mathbf{n}_x)} \tilde{\bar{u}}_j^{(\mathbf{n}_x)}\right) \tag{9.14}$$

and the Toosi-Larsson method proposes to minimize with respect to $\bar{\Delta}_{\mathbf{n}_x}$,$\bar{\Delta}_{\mathbf{n}_y}$, and $\bar{\Delta}_{\mathbf{n}_z}$ the error functional

$$e(\bar{\Delta}_{\mathbf{n}_x}, \bar{\Delta}_{\mathbf{n}_y}, \bar{\Delta}_{\mathbf{n}_z}) = \int_\Omega \left(\langle \widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_x)}, \widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_x)} \rangle + \langle \widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_y)}, \widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_y)} \rangle + \langle \widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_z)}, \widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_z)} \rangle \right)^{\frac{1}{2}} \tag{9.15}$$

in order to improve the mesh size in each direction.

The approach presented in this paper starts from an existing adaptation method for the numerical approximation errors for steady RANS flows, see [8] and the monograph [39]. In order to extend it to LES/hybrid, we try to combine (a) the existing adaptation of mesh for RANS or non-turbulent steady flow with (b) a special adaptation for LES error model built according to the Toosi-Larsson method. Focusing on hybrid modeling, we could hope (i) a rather good quasi-convergent capturing in RANS regions, to be combined with (ii) a suffficiently predictive resolution in LES region.

In this work, we focus on flows which are of a somewhat intermediate difficulty. These turbulent flows are assumed to be quasi-periodic with a rather well identified Strouhal number, and possibly quasi-steady in a large part of the computational domain. For a thin airfoil at small angle of attack for example, RANS calculation will produce a steady flow. VLES and hybrid RANS/LES calculations with medium meshes will produce a flow which is mainly steady, generally modeled with RANS, for which a deterministic adaptation criterion will be applied, but which presents also an unsteady region with vortices, where LES modeling applies, and for which a LES-based adaptation criterion must be applied.

2 Modeling

In this and the following section, we briefly recall concepts already defined in the thesis. Firstly, because this chapter is a first version of a paper in preparation.

2.1 Navier-Stokes model

The compressible Navier-Stokes equations for mass, momentum and energy conservation read :

$$\mathcal{NS}(W) = 0 \quad + \quad \text{Initial and boundary conditions} \tag{9.16}$$

where $\mathcal{NS}(W) = 0$ holds for :

$$\left\{ \begin{array}{l} \dfrac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{u}) = 0\,, \\ \dfrac{\partial(\rho \mathbf{u})}{\partial t} + \text{div}(\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p - \text{div}\, \mathcal{T} = 0\,, \\ \dfrac{\partial(\rho E)}{\partial t} + \text{div}((\rho E + p)\mathbf{u}) - \text{div}(\mathcal{T} \cdot \mathbf{u}) + \text{div}(\lambda \nabla T) = 0\,, \end{array} \right. \tag{9.17}$$

where ρ denotes the density (kg/m^3), $\mathbf{u}$ the velocity (m/s), E the total energy per mass ($m^2.s^{-2}$), p the pressure (N/m^2), given by : $p = (\gamma - 1)\Big(\rho E - \frac{1}{2}\rho|\mathbf{u}|^2\Big)$ with $\gamma = 1.4$, T the temperature (K) such that $\rho C_v T = E - \frac{1}{2}\rho(u^2 + v^2 + w^2)$, $\lambda = \mu C_p / Pr$ (C_v being the specific heat at constant volume, μ the dynamic viscosity, C_p the specific heat at constant pressure, Pr the Prandtl number). $\mathcal{T}$ is the laminar stress tensor :

$$\mathcal{T} = \mu \left[(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) - \frac{2}{3} \text{div}\, \mathbf{u}\, \mathbb{I}_3 \right] \,,$$

where (in 3D) $\mathbf{u} = (u_1, u_2, u_3)$, μ the laminar dynamic viscosity ($kg/(m.s)$) and λ the laminar thermal conductivity.

2.2 LES model

We have now to recall the LES analysis for the compressible model. For this we recall some notations. The filtered Navier-Stokes equations are considered. The density Favre filter $\widehat{f} = \overline{(\rho f)}/(\bar{\rho})$ (where the over-line denotes the gris filter) is applied and its solution is denoted $\bar{\rho}, \widehat{u}_1, \widehat{u}_2, \widehat{u}_3, \widehat{e}$.

The filtering of compressible Navier-Stokes equations gives the motion of large structures :

$$
\begin{cases}
\dfrac{\partial \bar{\rho}}{\partial t} + \dfrac{\partial(\bar{\rho}\widehat{u}_j)}{\partial x_j} = 0\,, \\[2ex]
\dfrac{\partial(\bar{\rho}\widehat{u}_i)}{\partial t} + \dfrac{\partial(\bar{\rho}\widehat{u}_i\widehat{u}_j)}{\partial x_j} = -\dfrac{\partial \bar{p}}{\partial x_i} + \dfrac{\partial(\mu\widehat{P}_{ij})}{\partial x_j} - \dfrac{\partial M_{ij}^{(1)}}{\partial x_j} + \dfrac{\partial M_{ij}^{(2)}}{\partial x_j}\,, \\[2ex]
\dfrac{\partial(\bar{\rho}\widehat{E})}{\partial t} + \dfrac{\partial[(\bar{\rho}\widehat{E}+\bar{p})\widehat{u}_j]}{\partial x_j} = \dfrac{\partial(\widehat{u}_j\widehat{\sigma}_{ij})}{\partial x_i} - \dfrac{\partial \widehat{q}_j}{\partial x_j} + \dfrac{\partial}{\partial x_j}\Big(E_j^{(1)} + E_j^{(2)} + E_j^{(3)}\Big)\,,
\end{cases}
$$

in which :

- $\widehat{q}$ is the resolved heat vector flux, $\widehat{q} = -k\,\nabla(\widehat{T})$ where k is the heat conductivity and $\widehat{T}$ is the Favre filtered temperature,

- $E_j^{(1)}, E_j^{(2)}, E_j^{(3)}$ are defined e.g. in [27], $E_j^{(2)}, E_j^{(3)}$ are negligible compared to $E_j^{(1)}$ and the effect of $E_j^{(1)}$ will be neglected in the sequel,

- $M_{ij}^{(1)}$ is defined from the filtering of the convective term in the moment equation :

$$
\overline{\frac{\partial(\rho u_i u_j)}{\partial x_j}} = \frac{\partial(\bar{\rho}\widehat{u}_i\widehat{u}_j)}{\partial x_j} + \frac{\partial M_{ij}^{(1)}}{\partial x_j} \quad \Rightarrow \quad M_{ij}^{(1)} = \overline{\rho u_i u_j} - \bar{\rho}\widehat{u}_i\widehat{u}_j. \tag{9.18}
$$

- $M_{ij}^{(2)}$ is defined by :

$$
M_{ij}^{(2)} = \overline{\mu P_{ij}} - \mu\widehat{P}_{ij} \quad \text{with} \quad P_{ij} = 2S_{ij} - \frac{2}{3}S_{kk}\delta_{ij} \tag{9.19}
$$

where $S_{ij} = \dfrac{1}{2}\left(\dfrac{\partial u_i}{\partial x_j} + \dfrac{\partial u_j}{\partial x_i}\right)$.

As a consequence the resolved strain tensor is denoted by :

$$
\widehat{S_{ij}} = \frac{1}{2}\left(\frac{\partial \widehat{u}_i}{\partial x_j} + \frac{\partial \widehat{u}_j}{\partial x_i}\right)
$$

and the viscous term is written

$$
\widehat{\sigma} = \mu\widehat{P_{ij}} \;\text{ with }\; \widehat{P_{ij}} = 2\widehat{S_{ij}} - \frac{2}{3}\widehat{S_{kk}}\delta_{ij}. \tag{9.20}
$$

$M_{ij}^{(2)}$ is negligible compared to $M_{ij}^{(1)}$. The isotropic part of $M_{ij}^{(1)}$, namely $\frac{1}{3}M_{kk}^{(1)}\delta_{ij}$, can be neglected, [42], and its deviatoric part is defined by :

$$
T_{ij} = M_{ij}^{(1)} - \frac{1}{3}M_{kk}^{(1)}\delta_{ij} = -\mu_{SGS}\widehat{P_{ij}} = -\mu_{SGS}\Big(2\widehat{S_{ij}} - \frac{2}{3}\widehat{S_{kk}}\delta_{ij}\Big).
$$

In our theoretical development, the turbulent viscosity is defined according to Smagorinsky model [124] :

$$\mu_{SGS} = \bar{\rho}(C_s\bar{\Delta})^2|\widehat{S}| \;\text{ with }\; |\widehat{S}| = \sqrt{2\widehat{S}_{ij}\widehat{S}_{ij}} \tag{9.21}$$

and the model stress tensor is analogous to the one introduced in (5) for the incompressible case :

$$\tau_{ij}^{\text{model}} = T_{ij} = -\bar{\rho}(C_s\bar{\Delta})^2|\widehat{S}|\widehat{P_{ij}}. \tag{9.22}$$

In order to define a Large Eddy Simulation (LES) model from the above one, we restrict to the usual definition of the grid size Δ_g (cf. (9.1)). We replace in the viscous term of moment equation the viscosity μ by the incremented viscosity $\mu + \mu_{SGS}$. We restrict to quasi-isothermal flows and then we do not introduce a model in the energy equation. Our LES model writes :

$$\mathcal{NS}(W) = \begin{pmatrix} 0 \\ \operatorname{div}\left(\mu_{SGS}\left[(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}) - \tfrac{2}{3}\operatorname{div}\mathbf{u}\,\mathbb{I}_3\right]\right) \\ 0 \end{pmatrix}. \tag{9.23}$$

2.3 Spalart-Allmaras model and DDES

Various forms of the Spalart-Allmaras (SA) turbulence model exist. The original Spalart-Allmaras one equation turbulence model writes [125] :

$$\begin{aligned} \frac{\partial\tilde{\nu}}{\partial t} + \mathbf{u}\cdot\nabla\tilde{\nu} &= c_{b1}[1 - f_{t2}]\tilde{S}\tilde{\nu} - \left[c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2 \\ &+ \frac{1}{\sigma}\left[\nabla\cdot((\nu+\tilde{\nu})\nabla\tilde{\nu}) + c_{b2}\|\nabla\tilde{\nu}\|^2\right] + f_{t1}\Delta\mathbf{u}^2 . \end{aligned}$$

where $\tilde{\nu}$ is the kinematic eddy turbulent viscosity. In this paper we consider the simplified formulation considering $f_{t1} = 0$ and $f_{t2} = 0$:

$$\frac{\partial\rho\tilde{\nu}}{\partial t} + \frac{\partial u_j\rho\tilde{\nu}}{\partial x_j} = \rho c_{b1}\tilde{S}\tilde{\nu} - \rho c_{w1}f_w\left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{\rho}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu+\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial\tilde{\nu}}{\partial x_i}\frac{\partial\tilde{\nu}}{\partial x_i}\right].$$

The turbulent eddy viscosity is computed from :

$$\mu_t = \rho\tilde{\nu}f_{v1}$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \quad\text{and}\quad \chi = \frac{\tilde{\nu}}{\nu} \quad\text{with}\quad \nu = \frac{\mu}{\rho}.$$

Additional definitions are given by the following equations :

$$\tilde{S} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \quad \text{where} \quad \Omega = \|\nabla \times \mathbf{u}\| \, .$$

The magnitude of the vorticity is computed from the vorticity tensor where each component is given by $\omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$ and $\Omega = \sqrt{2 \sum_{i,j=1..3} \omega_{ij}\omega_{ij}}$. Symbol d holds for the distance from the field point to the nearest wall and

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \, .$$

Notice that we have the following relations :

$$\chi f_{v1} = \frac{\tilde{\nu} f_{v1}}{\nu} = \frac{\nu_t}{\nu} \quad \Longrightarrow \quad 1 - \frac{\chi}{1 + \chi f_{v1}} = 1 - \frac{\tilde{\nu}}{\nu + \nu_t} \, .$$

The constants are

$$\begin{array}{llll} \sigma = \frac{2}{3} & c_{b1} = 0.1355 & c_{b2} = 0.622 & \kappa = 0.41 \\ c_{w1} = \frac{c_{b1}}{\kappa} + \frac{1 + c_{b2}}{\sigma} & c_{w2} = 0.3 & c_{w3} = 2 & c_{v1} = 7.1 \, . \end{array}$$

Finally, the function f_w is computed as :

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{1/6} \quad \text{with} \quad g = r + c_{w2}\left(r^6 - r\right) \quad \text{and} \quad r = \min\left(\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2} \, , \, 10\right) \, .$$

The standard SA one-equation model reads in pseudo-vector notations :

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} &= c_{b1}[1 - f_{t2}]\tilde{S}\tilde{\nu} - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right] \left(\frac{\tilde{\nu}}{d}\right)^2 \\ &+ \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}\right] \end{aligned}$$

where $f_{t2} = c_{t3} \exp\left(-c_{t4}\chi^4\right)$ with $c_{t3} = 1.2$ and $c_{t4} = 0.5$.

The DDES model [126] replaces the distance d by :

$$\tilde{d} = d - f_d \, max[0, (d - C_{DES}\Delta)]$$

in which $C_{DES} = 0.65$ and with ($\kappa = 0.41$)

$$f_d = 1 - tanh\left([8r_d]^3\right) \quad \text{where} \quad r_d = \frac{\nu + \nu_t}{\sqrt{\frac{\partial u_i}{\partial x_j}\frac{\partial u_i}{\partial x_j}}\kappa^2 d^2} .$$

$f_d = 0$ yields RANS for which we shall use a truncation error analysis while $r_d \ll 1$ yields the LES region for which we shall use the Germano-type error analysis due to Toosi and Larsson. The above DDES system is denoted in a compact way as :

$$\Psi_{\text{DDES}}(W) = 0 \tag{9.24}$$

where $W = (\rho, \rho u_1, \rho u_2, \rho u_3, \rho E, \rho\tilde{\nu})$ and

$$\Big(\Psi_{\text{DDES}}(W), \varphi\Big) = \int_0^T \int_\Omega \varphi \Big[W_t + \mathcal{S}_{\text{DDES}}(W) + \text{div}\left(\mathcal{F}_{\text{DDES}}(W)\right)\Big]\,\mathrm{d}\mathbf{x}\mathrm{d}t \tag{9.25}$$

$$+ \int_0^T \int_{\partial\Omega} \varphi \overline{\mathcal{F}}_{\text{DDES}}(W)\,\mathrm{d}\sigma\mathrm{d}t \tag{9.26}$$

where $\mathcal{S}_{\text{DDES}}(W)$, $\mathcal{F}_{\text{DDES}}(W)$, $\overline{\mathcal{F}}_{\text{DDES}}(W)$ hold respectively for the source term, the flux and the boundary flux of the DDES model.

3 Mesh-adaptation

In this first study, the focus is on flows which are essentially steady, with a rather small region of the computational domain in which we have a quasi periodic vortex shedding.

In the goal oriented option, we observe that the numerical error $\delta W = W - W_h$ on state variable W is solution of a linearised system

$$A\,\delta W_{\text{space-time}} = S_{\text{space-time}}$$

where :

- the linear operator A is the derivative of the Navier-Stokes (resp. URANS, or DDES) residual with respect to the state,
- the right-hand side $S_{\text{space-time}}$ of the system is the local error resulting from the space-time discretization (expressed in terms of the metric).

For minimizing the numerical error on the functional $j = (g, W)$ with respect to the mesh metric $\mathcal{M}$, it is sufficient to minimize the product of the adjoint state W^* with the right-hand side,

$$\delta j = (W^*, S_{\text{space-time}})$$

which will be made possible by the transformation of the local error into a quadratic function of the metric :

$$\min_{\mathcal{M}} \delta j = \min_{\mathcal{M}} (W^*, S_{\text{space-time}}(\mathcal{M}))$$

We obtain the minimum $\mathcal{M}_1$.

Similarly, assume that we have also a linear equation for the (test-filtered) error resulting from the LES modeling

$$A\,\delta W_{\text{LES}} = S_{\text{LES}}$$

where :

- the linear operator A is again the derivative of the Navier-Stokes residual with respect to the state,
- the right-hand side S_{LES} of the system, the local error of LES, is the divergence of the Germano expression.

As a result, the optimal mesh metric minimizing the total numerical *and* LES error on the functional writes :

$$\mathcal{M}_{\text{both}} = Arg \min_{\mathcal{M}} (W^*, S_{\text{space-time}}(\mathcal{M}) + S_{\text{LES}}(\mathcal{M})). \tag{9.27}$$

In this formulation, $S_{\text{space-time}}(\mathcal{M})$ and $S_{\text{LES}}(\mathcal{M})$ are not necessarily positive and therefore compensations of one error by the other one are possible and can be taken into account in the minimization.

However, up to now, we have not found a satisfying way for expressing explicitly S_{LES} as a function of $\mathcal{M}$ and therefore we cannot apply the strategy producing (9.27).

In this paper, we shall separate the error analysis into two steps, namely :

(a) research the RANS-optimal metric and,

(b) research of a somewhat LES-optimal metric by defining the test filter directions $(\mathbf{n}_k, k = 1, 3$ from the RANS-optimal metric and then follow the Toosi-Larsson for computing the optimal filter widths in these directions.

3.1 Riemannian metric

In the sequel, any mesh is represented by a Riemannian metric. See [8] and [39] for more details on this approach. The Riemannian metric $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is a symmetric positive matrix 3×3 field defined on the computational domain :

$$\mathcal{M} : \ \mathbf{x} \in \Omega \mapsto \mathcal{M}(\mathbf{x}) \ = \ \mathcal{R}(\mathbf{x})\, \Lambda(\mathbf{x})\, {}^t\mathcal{R}(\mathbf{x}), \tag{9.28}$$

where diagonal matrix $\Lambda(\mathbf{x})$ is

$$\begin{pmatrix} \lambda_1(\mathbf{x}) & & \\ & \lambda_2(\mathbf{x}) & \\ & & \lambda_3(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} h_1^{-2}(\mathbf{x}) & & \\ & h_2^{-2}(\mathbf{x}) & \\ & & h_3^{-2}(\mathbf{x}) \end{pmatrix}. \tag{9.29}$$

$\mathcal{R}(\mathbf{x})$ is an orthonormal matrix providing the local orientation of mesh stretching through the eigenvectors $(\mathbf{v}_i(\mathbf{x}))_{i=1,3}$, $(\lambda_i(\mathbf{x}))_{i=1,3}$ are the local eigenvalues. $(h_i(\mathbf{x}))_{i=1,3} = (\lambda_i(\mathbf{x})^{-\frac{1}{2}})_{i=1,3}$ are the local mesh sizes along the *principal directions* $\mathbf{p}_{1,\mathcal{M}}, \mathbf{p}_{2,\mathcal{M}}, \mathbf{p}_{3,\mathcal{M}}$ of $\mathcal{M}$ defined by :

$$\mathbf{p}_{k,\mathcal{M}}(\mathbf{x}) \ = \ \mathcal{R}(\mathbf{x})\, \mathbf{e}_k\, {}^t\mathcal{R}(\mathbf{x}) \tag{9.30}$$

where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ are the three Cartesian unitary vectors in x, y, z directions. The density d of $\mathcal{M}$ is defined from its eigenvalues as

$$d(\mathbf{x}) = \det\Big(\mathcal{M}(\mathbf{x})\Big)^{\frac{1}{2}} = (\lambda_1(\mathbf{x})\, \lambda_2(\mathbf{x})\, \lambda_3(\mathbf{x}))^{\frac{1}{2}} = (h_1(\mathbf{x})\, h_2(\mathbf{x})\, h_3(\mathbf{x}))^{-1}.$$

We decompose $\mathcal{M}$ as follows :

$$\mathcal{M}(\mathbf{x}) = d^{\frac{2}{3}}(\mathbf{x})\,\mathcal{R}(\mathbf{x}) \begin{pmatrix} r_1^{-\frac{2}{3}}(\mathbf{x}) & & \\ & r_2^{-\frac{2}{3}}(\mathbf{x}) & \\ & & r_3^{-\frac{2}{3}}(\mathbf{x}) \end{pmatrix} {}^t\mathcal{R}(\mathbf{x})$$

where the r_i's define the stretching strength and where the density d controls the local level of accuracy of $\mathcal{M}$. The *complexity* $\mathcal{C}$ of $\mathcal{M}$ is defined by :

$$\mathcal{C}(\mathcal{M}) = \int_\Omega d(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}.$$

This real-value parameter quantifies the global level of accuracy of $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\mathbf{\Omega}}$.

A discrete tetrahedrization $\mathcal{H}$ is *a unit mesh* for the metric $\mathcal{M}$ if any of its edges $\mathbf{ab}$ has a length in the metric sufficiently close to unity :

$$\frac{1}{\sqrt{2}} \leq \int_0^1 \sqrt{{}^t\mathbf{ab}\;\mathcal{M}(\mathbf{a}+t\,\mathbf{ab})\;\mathbf{ab}}\;\mathrm{d}t \leq \sqrt{2}$$

Then the complexity can also be interpreted as the continuous counterpart of the *number of vertices of a discrete unit mesh* while d can be interpreted as the continuous counterpart of the *number of vertices per volume unit of a discrete unit mesh.*

Lastly, we call *refinement* the process which replaces a unit mesh of a given metric $\mathcal{M}$ with local mesh size $(h_1(\mathbf{x})\,h_2(\mathbf{x})\,h_3(\mathbf{x}))$ by a unit mesh of $\mathcal{M}/\beta^2$ with local mesh size $(\beta h_1(\mathbf{x})\,\beta h_2(\mathbf{x})\,\beta h_3(\mathbf{x}))$ and complexity $\mathcal{C}(\mathcal{M}/\beta^2) = \beta^3\mathcal{C}(\mathcal{M})$ where refinement factor β is smaller than one [2] .

3.2 Adaptation sensor for compressible flow

3.2.1 Feature-based adaptation sensor

In the case of a steady RANS calculation of a compressible flow, an efficient approach is to minimize the L^4 interpolation error on the Mach number [8][39]. The local interpolation error $e_{\mathcal{M}}$ (9.31) is evaluated in terms of the Hessian H_M of Mach number M and of the metric $\mathcal{M}$ used for generating the mesh :

$$e_{\mathcal{M}}(\mathbf{x}) = (M - \pi_{\mathcal{M}}M)(\mathbf{x}) \;=\; \frac{1}{10}\,\mathrm{trace}\Big(\mathcal{M}(\mathbf{x})^{-\frac{1}{2}}\,|H_M(\mathbf{x})|\,\mathcal{M}(\mathbf{x})^{-\frac{1}{2}}\Big), \qquad (9.31)$$

in which $\mathbf{x} \in \Omega$, $(\mathbf{v}_i)_{i=1,3}$ are the local eigen-directions of $\mathcal{M}$, and $(h_i)_{i=1,3}$ are the local sizes of $\mathcal{M}$ along these directions. This local error [3] is a spatially second-order error. In

---

[2] For $\beta = 2$ this refinement is equivalent to dividing mesh size by a factor 2 and multiplying the number of vertices in 3D by a factor 8.

[3] $e_{\mathcal{M}}$ is an *a priori* error when we consider that M is the exact Mach number field. In practice, it will be an *a posteriori* error since $e_{\mathcal{M}}$ will be computed from a discrete solution through a recovery technique.

order to take into account boundary layers in good conditions, it has been observed (see again [8] [39]) that the error norm to be minimized is an L^4 norm. The mesh adaptation problem is then written :

$$\text{Find } \mathcal{M}_1 = \min_{\mathcal{M}} \int_\Omega \Big(\text{trace}\Big(\mathcal{M}(\mathbf{x})^{-\frac{1}{2}}\, |H_M(\mathbf{x})|\, \mathcal{M}(\mathbf{x})^{-\frac{1}{2}}\Big)\Big)^{\frac{1}{4}}\, \mathrm{d}\mathbf{x} \tag{9.32}$$

under the constraint that the complexity, or integral of the metric density is equal to a specified number $\mathcal{N}$:

$$\mathcal{C}(\mathcal{M}) = \int_\Omega \det(\mathcal{M})\mathrm{d}\Omega = \mathcal{N}.$$

Expressing via (9.31) the functional (9.32) in terms of H_M and assuming that this Hessian is sufficiently smooth, the solution of this constrained optimisation problem can be explicitly computed [8][39] :

$$\mathcal{M}_1 = D_1\ \det(|H_M|)^{\frac{-1}{11}}\, |H_M|, \text{ with } D_1 = N^{\frac{2}{3}} \left(\int_\Omega \det(|H_M|)^{\frac{4}{11}} \mathrm{d}\mathbf{x}\right)^{-\frac{2}{3}}. \tag{9.33}$$

3.2.2 Goal-oriented adaptation sensor

We consider the Goal Oriented unsteady formulation as introduced in [23] and its extension to RANS as in [22, 39]. We keep the notations of these papers. We want to minimize the error $(g, W - W_h)$ committed in the approximation of the functional (or scalar output) :

$$j = (g, W),$$

where W is the exact solution of the state equation (9.24,9.26) and W_h the approximate solution. Let us introduce the adjoint state W^*, solution of the adjoint system :

$$\left(\frac{\partial \Psi}{\partial W}\right)^* W^* = g. \tag{9.34}$$

We reproduce now in short the error estimate developed in [84],[23],[22][9]. The functional error estimates writes :

$$|(g, W_h - W)| \approx \mathcal{E}^p_{\text{space}}(\mathcal{M}) \tag{9.35}$$

with

$$\begin{aligned}
\mathcal{E}^p_{\text{space}}(\mathcal{M}) \approx & \int_0^T \int_\Omega |W^*| \left|\Big(W - \pi_{\mathcal{M}} W\Big)_t + \mathcal{S}_{\text{DDES}}(W) - \pi_{\mathcal{M}} \mathcal{S}(W)_{\text{DDES}}\Big)\right| \mathrm{d}\mathbf{x}\, \mathrm{d}t \\
& + \int_0^T \int_\Omega \left|\frac{\partial \mathcal{F}_{\text{DDES}}}{\partial W} \cdot \nabla W^*\right|\ |W - \pi_{\mathcal{M}} W| \mathrm{d}\mathbf{x}\, \mathrm{d}t \\
& + \int_0^T \int_\Gamma |W^*|\ \left|(\bar{\mathcal{F}}_{\text{DDES}}(W) - \pi_{\mathcal{M}} \bar{\mathcal{F}}_{\text{DDES}}(W)) \cdot \mathbf{n}\right| \mathrm{d}\Gamma\, \mathrm{d}t.
\end{aligned}$$

Neglecting the boundary term, we get :

$$\mathcal{E}_{\text{space}}(\mathcal{M}) \;\approx\; \int_0^T \int_\Omega \text{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\,\mathbf{H}(\mathbf{x},t)\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\right)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t, \tag{9.36}$$

$$\text{with}\quad \mathbf{H}(\mathbf{x},t) = \left|W_t^* + W^*\frac{\partial \mathcal{S}}{\partial W} + \frac{\partial \mathcal{F}}{\partial W}\cdot\nabla W^*\right|\;|H(W)|\,, \tag{9.37}$$

where $H(W)$ is the Hessian of W.

Lemma 3.1 Unified numerical error criterion. *The two numerical error criteria, namely (9.31,9.32,9.33) for the feature-based option and (9.35,9.36,9.37) for the Goal-Oriented option, examined previously, and the related optimization problems can be unified as follows :*

$$\begin{aligned}
&\mathcal{M}_{num}(\mathbf{x}) = \mathcal{R}_{num}(\mathbf{x})\begin{bmatrix} h_{1,num}(\mathbf{x}) & & \\ & h_{2,num}(\mathbf{x}) & \\ & & h_{3,num}(\mathbf{x})\end{bmatrix}{}^t\mathcal{R}_{num}(\mathbf{x})\\
&\mathcal{M}_{num} = \mathit{Argmin}\;\mathcal{E}(\mathcal{M}) = \mathit{Argmin}\int_0^T \mathcal{E}_{\text{space},t}(\mathcal{M},t)\,\mathrm{d}t \quad \mathit{with}\\
&\mathcal{E}_{\text{space},t}(\mathcal{M},t) = \int_\Omega \left[\text{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\,\mathbf{H}(\mathbf{x},t)\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\right)\right]^p \mathrm{d}x,
\end{aligned} \tag{9.38}$$

where :

- *in the Feature-Based case, a sensor M is computed from W, setting $\mathbf{H} = H_M$.*
- *in the Goal-Oriented case, $p = 1$, and W^* is the adjoint state, $\mathbf{H}$ is defined as in (9.37).*

Since the flow of interest is quasi-steady in a large part of the computational domain, we keep the metric $\mathcal{M}_{\text{num}}$ resulting from the optimization of the numerical error functional $\mathcal{E}(\mathcal{M})$ as numerical criterion of the proposed extension to DDES.

3.3 Toosi-Larsson adaptation step

A simplified way to explain the Toosi-Larsson method [130] is to consider :

- an anisotropic Cartesian mesh available with Cartesian directions $\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z$,
- three test filters evaluated on the flow in these three directions,
- then new local mesh sizes in these directions $\Delta x(\mathbf{x}, \mathbf{n}_x)$, $\Delta y(\mathbf{x}, \mathbf{n}_x)$, $\Delta z(\mathbf{x}, \mathbf{n}_x)$, $\mathbf{x} \in \Omega$, are defined, defining a new adapted mesh, with variation of mesh size and stretching, but with the same principal directions.

We propose here an equivalent approach for the case of a metric-based anisotropic mesh. Let $\mathcal{M}_{\text{num}}$ be *one of the two* metrics defined in (9.38) with local mesh sizes $(h_{1,\text{num}}(\mathbf{x})$, $h_{2,\text{num}}(\mathbf{x})$, $h_{3,\text{num}}(\mathbf{x}))$ according to (9.29) and $\mathcal{H}_{\text{num}}$ a unit mesh of $\mathcal{M}_{\text{num}}$. The LES or hybrid simulation is performed on $\mathcal{H}_{\text{num}}$ with a filter size :

$$\bar{\Delta}_0(\mathbf{x}) = (h_{1,\text{num}}(\mathbf{x})\, h_{2,\text{num}}(\mathbf{x})\, h_{3,\text{num}}(\mathbf{x}))^{\frac{1}{3}}.$$

Then three test filters are built in principal directions $(\mathbf{p}_k(\mathbf{x})),\ k = 1, 3$ of $\mathcal{M}_{\text{num}}$ (according to the definition in (9.30)),

$$\mathbf{n}_k(\mathbf{x}) = \widetilde{\Delta}_{\mathbf{n}_k}(\mathbf{x}) \mathbf{p}_{k,\mathcal{M}_0}(\mathbf{x}) \tag{9.39}$$

having the length

$$||\mathbf{n}_k(\mathbf{x})|| = \widetilde{\Delta}_{\mathbf{n}_k}(\mathbf{x}) = 2\bar{\Delta}(\mathbf{x}) \quad k = 1, 3. \tag{9.40}$$

The test filter width $= \widetilde{\Delta}_{\mathbf{n}_k}$ will vary in the neighbohood of a reference width :

$$||\mathbf{n}_{k,0}(\mathbf{x})|| = \widetilde{\Delta}_{\mathbf{n}_{k,0}}(\mathbf{x}) = 2\bar{\Delta}_0(\mathbf{x}) \quad k = 1, 3. \tag{9.41}$$

We consider the effect of the test filter, which we denote $\tilde{\cdot}^{(\mathbf{n}_k)}$. The error source term (9.14) becomes in the compressible case :

$$\widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_k)}(\mathbf{x}) = \frac{\partial}{\partial x_j} \Big(\tau_{ij}^{\text{model}}(\widetilde{\overline{u}}^{(\mathbf{n}_k)}) - \widetilde{\tau_{ij}^{\text{model}}(\overline{u})}^{(\mathbf{n}_k)} - \mathcal{M}_{ij} \Big). \tag{9.42}$$

where

$$\mathcal{M}_{ij} = \widetilde{\overline{\rho u_i u_j}}^{(\mathbf{n}_k)} - \Big(\frac{\widetilde{\overline{\rho u_i}}^{(\mathbf{n}_k)} \widetilde{\overline{\rho u_j}}^{(\mathbf{n}_k)}}{\widetilde{\overline{\rho}}^{(\mathbf{n}_k)}} \Big) \tag{9.43}$$

It can be shown [51] that the SGS and the sub-test stress tensors are related by the following identity :

$$\mathcal{L}_{ij} = \widetilde{\bar{\rho}\tilde{u}_i\tilde{u}_j}^{(\mathbf{n}_k)} - \frac{1}{\widetilde{\overline{\rho}}^{(\mathbf{n}_k)}} \Big(\widetilde{\bar{\rho}\tilde{u}_i}^{(\mathbf{n}_k)} \widetilde{\bar{\rho}\tilde{u}_j}^{(\mathbf{n}_k)} \Big) = \mathcal{M}_{ij} - \widetilde{M_{ij}^{(1)}}^{(\mathbf{n}_k)}. \tag{9.44}$$

Let us denote

$$L_{ij} = \mathcal{L}_{ij} - \frac{1}{3}\mathcal{L}_{kk}\delta_{ij}. \tag{9.45}$$

Then, to each test filter $\mathbf{n}_k$ corresponds an error source term

$$\widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_k)}(\mathbf{x}) = \frac{\partial}{\partial x_j}\Big(\tau_{ij}^{\text{model}}(\widetilde{\overline{u}}^{(\mathbf{n}_k)}) - \widetilde{\tau_{ij}^{\text{model}}(\overline{u})}^{(\mathbf{n}_k)} - L_{ij} \Big). \tag{9.46}$$

From (9.44),(9.45) and (9.46), we deduce :

Lemma 3.2

$$\begin{aligned} \widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_k)}(\mathbf{x}) \;\; = \;\; & \frac{\partial}{\partial x_j}\Big(\widetilde{(C\bar{\Delta}^2)\bar{\rho}|\widehat{S}|\widehat{P}_{ij}}^{(\mathbf{n}_k)} - (C\bar{\Delta}^2)\Big(\frac{\widetilde{\Delta}^{(\mathbf{n}_k)}}{\bar{\Delta}}\Big)^2 \widetilde{\overline{\rho}}^{(\mathbf{n}_k)} |\widetilde{\widehat{S}}^{(\mathbf{n}_k)}| \, \widetilde{\widehat{P}_{ij}}^{(\mathbf{n}_k)} \\ & - \widetilde{\bar{\rho}\widehat{u}_i\widehat{u}_j}^{(\mathbf{n}_k)} + \tfrac{1}{\widetilde{\overline{\rho}}^{(\mathbf{n}_k)}} (\widetilde{\bar{\rho}\widehat{u}_i}^{(\mathbf{n}_k)} \widetilde{\bar{\rho}\widehat{u}_j}^{(\mathbf{n}_k)}) + \tfrac{1}{3}\Big(\widetilde{\bar{\rho}\widehat{u}_l\widehat{u}_l}^{(\mathbf{n}_k)} - \tfrac{1}{\widetilde{\overline{\rho}}^{(\mathbf{n}_k)}} (\widetilde{\bar{\rho}\widehat{u}_l}^{(\mathbf{n}_k)} \widetilde{\bar{\rho}\widehat{u}_l}^{(\mathbf{n}_k)}) \Big)\delta_{ij} \Big) \end{aligned} \tag{9.47}$$

$$i = 1, \dots, 3, \;\; k = 1, \dots, 3 \;\; \square$$

In the Piomelli dynamic formulation, C (or $C\bar{\Delta}^2$) is the unknown scalar parameter to be choosen at each point of the computational domain in order to optimize the efficiency of the LES model. In the present analysis, $C\bar{\Delta}^2$ is known, and the optimization variables are the three filter sizes $\widetilde{\Delta}^{(\mathbf{n}_1)}, \widetilde{\Delta}^{(\mathbf{n}_2)}, \widetilde{\Delta}^{(\mathbf{n}_3)}$. We observe that, when we freeze all the filtered terms, $\widetilde{\Delta}^{(\mathbf{n}_k)}$ appears explicitly in (9.47) under the form of a quadratic term. But for all "hat" filtered quantity, we remark that they approximatively differ from the non-filtered quantity by a rest proportional to the product of square of filter width times a second derivative of unfiltered quantity. However, following [130], and for the sake of simplicity, we work in the neighborhood of a LES filter width $\bar{\Delta}_0$:

$$\widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_{k,0})} = \widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_k)}\big|_{\bar{\Delta}=\bar{\Delta}_0,\; \widetilde{\Delta}^{(\mathbf{n}_k)}=\widetilde{\Delta}^{(\mathbf{n}_{k,0})}=2\bar{\Delta}_0}, \quad i = 1, ..., 3, \;\; k = 1, ..., 3. \tag{9.48}$$

In the neighborhood of $\bar{\Delta} = \bar{\Delta}_0,\; \widetilde{\Delta}^{(\mathbf{n}_k)} = 2\bar{\Delta}_0$, we approximate the variation of $\widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_k)}$ in the neighborhood of $\mathbf{n}_{k,0}$ with the following quadratic formula :

$$\widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_k)} \;\; \approx \;\; \Big(\frac{\widetilde{\Delta}^{(\mathbf{n}_k)}}{2\,\bar{\Delta}_0}\Big)^2 \widetilde{\overline{\mathcal{F}}}_i^{(\mathbf{n}_{k,0})} \quad i = 1, ..., 3, \;\; k = 1, ..., 3, \tag{9.49}$$

We observe that this approximation of an error term for LES is coherent with the fact that the LES model is obtained by perturbating the Navier-Stokes model by the SGS term which is of second order with respect to the mesh size $\bar{\Delta}$. We introduce the notations :

$$\Delta_k = \frac{\widetilde{\Delta}^{(\mathbf{n}_k)}}{2} \;\; ; \;\; \mathcal{G}_k = \bar{\Delta}^{-2}\Big(\langle \tilde{\tilde{\mathcal{F}}}^{(\mathbf{n}_k,0)}, \tilde{\tilde{\mathcal{F}}}^{(\mathbf{n}_k,0)} \rangle\Big)^{\frac{1}{2}} \quad k = 1, ..., 3, \tag{9.50}$$

where the scalar product $\langle a, b\rangle = \sum_{i=1,3} a_i b_i$. We shall now find the following metric (aligned with numerical metric $\mathcal{M}_{\text{num}}$ defined in (9.38)) :

$$\mathcal{M}(\mathbf{x}) = \mathcal{R}_{\text{num}}(\mathbf{x}) \begin{pmatrix} (\frac{h_{1,\text{num}}}{\bar{\Delta}}\Delta_1)^{-2}(\mathbf{x}) & & \\ & (\frac{h_{2,\text{num}}}{\bar{\Delta}}\Delta_2)^{-2}(\mathbf{x}) & \\ & & (\frac{h_{3,\text{num}}}{\bar{\Delta}}\Delta_3)^{-2}(\mathbf{x}) \end{pmatrix} {}^t\mathcal{R}_{\text{num}}(\mathbf{x})$$

which minimizes the functional :

$$e(\Delta_1, \Delta_2, \Delta_3) = \int_\Omega \Big(\mathcal{G}_1^2\Delta_1^4 + \mathcal{G}_2^2\Delta_2^4 + \mathcal{G}_3^2\Delta_3^4\Big)^{\frac{1}{2}} \mathrm{d}\mathbf{x} \tag{9.51}$$

under the constraint :

$$\mathcal{C}(\mathcal{M}) = \mathcal{N},$$

where N is prescribed by the user. We observe that

$$\mathcal{C}(\mathcal{M}) = \int_\Omega \bar{\Delta}^3(h_1h_2h_3\Delta_1\Delta_2\Delta_3)^{-1}\mathrm{d}\mathbf{x},$$

which reduces the problem to a minimization with respect to $(\Delta_1, \Delta_2, \Delta_3)$ under the constraint that the integral of their product is specified. Let us denote by $(\Delta_1^{\text{opt}}, \Delta_2^{\text{opt}}, \Delta_3^{\text{opt}})$ the solution of the minimization problem and :

$$\zeta^{\text{opt}} = \Big(\Delta_1^{\text{opt}}\Delta_2^{\text{opt}}\Delta_3^{\text{opt}}\Big)^{-1}.$$

In any point $\mathbf{x}$, knowing solely ζ^{opt}, we can deduce the Δ_k^{opt}'s since the (squared) value of the integrand

$$g(\mathbf{x}) = \mathcal{G}_1^2(\mathbf{x})\Delta_1^4(\mathbf{x}) + \mathcal{G}_2^2(\mathbf{x})\Delta_2^4(\mathbf{x}) + \mathcal{G}_3^2(\mathbf{x})\Delta_3^4(\mathbf{x})$$

must be minimized with respect to the $((\Delta_1^{\text{opt}})^4, (\Delta_2^{\text{opt}})^4, (\Delta_3^{\text{opt}})^4)$, with the constraint that their product $(\zeta^{\text{opt}})^{-4}$ is given. This implies that

$$\begin{aligned} \mathcal{G}_1^2(\mathbf{x})(\Delta_1^{\text{opt}})^4(\mathbf{x}) &= \mathcal{G}_2^2(\mathbf{x})(\Delta_2^{\text{opt}})^4(\mathbf{x}) = \mathcal{G}_3^2(\mathbf{x})(\Delta_3^{\text{opt}})^4(\mathbf{x}) \\ &= \Big((\zeta^{\text{opt}})^{-4}\mathcal{G}_1^2(\mathbf{x})\mathcal{G}_2^2(\mathbf{x})\mathcal{G}_3^2(\mathbf{x})\Big)^{\frac{1}{3}}, \end{aligned} \tag{9.52}$$

or, for $k = 1, 3$:

$$\Delta_k^{\text{opt}}(\mathbf{x}) = (\zeta^{\text{opt}})^{-\frac{1}{3}} \Big(\mathcal{G}_1(\mathbf{x})\mathcal{G}_2(\mathbf{x})\mathcal{G}_3(\mathbf{x})\Big)^{\frac{1}{6}} \Big(\mathcal{G}_k(\mathbf{x})\Big)^{-\frac{1}{2}}.$$

It remains to minimize

$$\mathcal{K}(\zeta) = \int_\Omega K(\mathbf{x})\zeta^{-\frac{2}{3}}(\mathbf{x})\mathrm{d}\mathbf{x},$$

with

$$K(\mathbf{x}) = 3\Big(\mathcal{G}_1(\mathbf{x})\mathcal{G}_2(\mathbf{x})\mathcal{G}_3(\mathbf{x})\Big)^{\frac{1}{3}},$$

under the constraint $\int_\Omega \zeta(\mathbf{x})\mathrm{d}\mathbf{x} = \mathcal{N}$. The resolution of this minimization problem gives :

$$K(\mathbf{x})\zeta^{-\frac{5}{3}}(\mathbf{x}) = \mathit{const.} \Rightarrow \zeta(\mathbf{x}) = \mathit{const.}^{-\frac{3}{5}} K(\mathbf{x})^{\frac{3}{5}},$$

$$\mathcal{N} = \int_\Omega \zeta \,\mathrm{d}\mathbf{x} = \mathit{const.}^{-\frac{3}{5}} \int_\Omega K^{\frac{3}{5}} \,\mathrm{d}\mathbf{x} \Rightarrow \mathit{const.}^{-\frac{3}{5}} = \mathcal{N}\Big(\int_\Omega K^{\frac{3}{5}} \,\mathrm{d}\mathbf{x}\Big)^{-1},$$

$$\zeta^{\text{opt}}(\mathbf{x}) = \Big(\int_\Omega K^{\frac{3}{5}}(\mathbf{x}')\mathrm{d}x'\Big)^{-1} K(\mathbf{x})^{\frac{3}{5}}\, \mathcal{N}, \tag{9.53}$$

and

$$\Delta_k^{\text{opt}}(\mathbf{x}) = \Big(\mathcal{G}_1(\mathbf{x})\mathcal{G}_2(\mathbf{x})\mathcal{G}_3(\mathbf{x})\Big)^{\frac{1}{6}} \Big(\mathcal{G}_k(\mathbf{x})\Big)^{-\frac{1}{2}} \Big(\int_\Omega K^{\frac{3}{5}}(\mathbf{x}')\mathrm{d}\mathbf{x}'\Big)^{\frac{1}{3}} K(\mathbf{x})^{-\frac{1}{5}}\, \mathcal{N}^{-\frac{1}{3}}. \tag{9.54}$$

In practice we think it is better that the initial metric $\mathcal{M}_0$ defining the filter should be the metric defining the previous mesh while the test filter directions will be chosen from the novel mean flow adaptation metric $\mathcal{M}_1$. then the metric adapting to both Mach field and LES would write :

$$\mathcal{M}_{\text{H,DDES}}(\mathbf{x}) = \mathcal{R}_1(\mathbf{x}) \begin{pmatrix} (\frac{h_1^1\Delta_1}{\overline{\Delta}})^{-2}(\mathbf{x}) & & \\ & (\frac{h_2^1\Delta_2}{\overline{\Delta}})^{-2}(\mathbf{x}) & \\ & & (\frac{h_3^1\Delta_3}{\overline{\Delta}})^{-2}(\mathbf{x}) \end{pmatrix} {}^t\mathcal{R}_1(\mathbf{x}).$$

This step will probably increase the complexity $\mathcal{C}(\mathcal{M}_{\text{H,DDES}})$ which will be larger than N. A final step of renormalization is then necessary :

$$\mathcal{M}_{\text{H,DDES}}^{\text{new}} = N^{\frac{2}{3}} \Big(\mathcal{C}(\mathcal{M}_{\text{H,DDES}})\Big)^{-\frac{2}{3}} \mathcal{M}_{\text{H,DDES}}.$$

4 Space and time approximation

In the previous sections, space and time discretizations are not considered. Indeed, a continuous optimization problem is introduced and solved under the form of an optimality system. It is this optimality system which is finally discretized by replacing the direct dependancy of the flow field to the metric by a chain of mapping :

$$\mathcal{M} \mapsto \mathcal{H} \mapsto W_{\mathcal{H}} \tag{9.55}$$

from the metric $\mathcal{M}$ to the discrete flow field $W_{\mathcal{H}}$ via the construction of a unit mesh $\mathcal{H}$ of the metric.

4.1 CFD numerics

The software used in this paper is the NiceFlow software and relies on the *vertex-centered* MUSCL approximation for *tetrahedrizations* described in details in Chapter 2. As the software Wolf used in Chapter 8, NiceFlow involves mesh adaptation functionalities described in the previous chapter, also described in [39] : computation of metric-based adaptation criteria, regeneration of adapted anisotropic meshes.

4.2 Implementation of LES criterion

4.2.1 LES filter

The SGS term is assembled by element and the filter size $\bar{\Delta} = (h_1 h_2 h_3)^{\frac{1}{3}}$ is evaluated from the volume of the local element jt :

$$\bar{\Delta}_{|jt} = \Big(vol(jt)\Big)^{\frac{1}{3}}.$$

4.2.2 Test filter

In [?] an isotropic test filter is built at each vertex is from a mean on the elements jt which have is as vertex, which we write $jt \ni is$:

$$\widetilde{w(is)} = \Big(\sum_{jt \ni is} \sum_{js \in jt} vol(jt)\Big)^{-1} \sum_{jt \ni is} \sum_{js \in jt} vol(jt)\, w(js) \tag{9.56}$$

where the $\sum_{js \in jt}$ means the sum over the vertices js of element jt and $vol(jt)$ holds for the volume of element jt. It results that the integration area of the test filter is the P_1 basis function support, or union of elements around a vertex. Equivalently, the test filter is a mean on the set of neighboring elements, of volume $Vol_{is} = \sum_{jt \ni is} vol(jt)$. Defining as N_{is} the number of elements having vertex is as a vertex, and assuming that the volume of elements around vertex is is close to $vol(jt)$ for a element containing is, then $Vol_{is} \approx N_{is} vol(jt)$. Taking $vol(jt)$ as the integration area of the model filter, we can set :

$$\frac{\widetilde{\Delta}}{\Delta} = N_{is}^{\frac{1}{3}}.$$

For building the anisotropic test filter directions, we use the principal directions $\mathbf{p}_{k,\mathcal{M}}$ (see (9.30)) of the background metric :

$$\mathbf{n}_k = 2\bar{\Delta}\ \mathbf{p}_{k,\mathcal{M}}.$$

The anisotropic filter according to vector $\mathbf{n}_k$, that is, aligned with the unit vector $\mathbf{p}_{k,\mathcal{M}}$, and of filter width $2\bar{\Delta}$ writes :

$$\widetilde{w(is)}^{k} = \sum_{jt \ni is} \sum_{js \in jt} vol(jt) |\langle \frac{\mathbf{isjs}}{|isjs|}, \mathbf{n}_k \rangle|\ w(js) \Big(\sum_{jt \ni is} \sum_{js \in jt} vol(jt) |\langle \frac{\mathbf{isjs}}{|isjs|}, \mathbf{n}_k \rangle| \Big)^{-1} \tag{9.57}$$

where **isjs** is the vector from vertex is to vertex js.

5 Adaptation algorithm

The adaptation algorithm solves the discretization of the optimality conditions for the optimal metric.

The flows under study are unsteady. In order to apply mesh adaptation, we shall use a version of the Transient Fixed Point introduced in [10]. A sketch of this algorithm is given by Algorithm 9.1.

Algorithm 9.1 – Transient Fixed Point

```
Given a complexity $N_{\text{prescribed}}$, an initial metric, $\mathcal{M}_0$ of complexity $c_0$, build a unit mesh $\mathcal{H}_0$ from $\mathcal{M}_0$

For $i_{adapt} = 0, n_{adapt}$
    • Compute over $[0,T]$ the flow $W_{i_{adapt}}$ from with mesh $\mathcal{H}_{i_{adapt}}$
    • Compute the $k_{\max}$ new metrics $\mathcal{M}^k_{i_{adapt}+1}$ of complexity $N_{\text{prescribed}}$ each taking into account the flow over $[t_k, t_{k+1}]$.
    • Compute the $k_{\max}$ new meshes $\mathcal{H}^k_{i_{adapt}+1}$ from $\mathcal{M}^k_{i_{adapt}+1}$
    • $i_{adapt} = i_{adapt} + 1$
End for $i_{adapt}$
```

Let us assume that Algorithm 9.1 iteratively converges (when i_{adapt} increases to infinity) to a fixed point $(W_\infty, \mathcal{M}_\infty)$. Then this fixed point is a numerical flow computed on a succession of meshes $\mathcal{M}_{\infty}^{k}, k = 1, k_{\text{max}}$, each mesh $\mathcal{H}_{\infty}^{k}$ being adapted to the best approximation (in some sense) of the flow on time interval $[t_k, t_{k+1}]$. Further, the sum of the complexities of the different meshes for $k = 1, k_{\text{max}}$ is the global complexity $k_{\text{max}}\mathcal{N}_{\text{prescribed}}$.

As in the previous chapter, since we are interested by vortex shedding flows past blunt bodies, we shall work with the *single-mesh option of the transient fixed point.* Indeed, we want to work with only one mesh for the whole time interval, and we put :

$$k_{\text{max}} = 1\ . \tag{9.58}$$

6 Numerical applications

In this first application of the proposed mesh adaptation method, we concentrate on the flow around a circular cylinder, and in particular on two typical flows, depending of their Reynolds number :
- a typical subcritical flow is the flow at Reynolds number 3900 in which the boundary layer is said a laminar one, while the detached wake is turbulent, we also consider the case of the flow at Reynolds number 20K,
- a typical supercritical flow if Reynolds 1 Million, in which boundary layers and wake pewsent turbulent characteristic.

Here subcritical and supercritical refer to the drag crisis Reynolds numbers around 300000 at which the drag decreases rapidly.

6.1 Subcritical flow calculation with adaptation

The subcritical cases are flows around a circular cylinder at Reynolds number 3900 and $20K$. Several computations and measurements are described in Chapter 6. The boundary layer being of laminar regime (although participating to the mechanism of the turbulent global flow), the state-of-art modelisation for this flow is to apply a LES model of medium sophistication. We choose the combination of the VMS formulation with the WALE SGS model, as described in the chapter recalling some turbulence models.

A first mesh-adaptive calculation uses the Hessian of Mach number as anisotropic adaptation criterion, with the L^4 norm, according to the results of available experiments, see for example [**?**]. For the same reasons as for the space-time adaption study of Chapter 8, this unsteady flow is adapted with a mesh unchanged during the time interval , using the single-mesh transient fixed point.

6.1.1 Re = 3900

First, let us take a look at Figure 9.1 to see what our algorithm produces for $\Delta_1^{\text{opt}}/\bar{\Delta}$. Note that this value varies enormously, from 10^{-4} to 86. Keeping in mind that this value multiplies the local mesh size h_1 we decide to reduce the range of possible values for $\Delta_1^{\text{opt}}/\bar{\Delta}$ (Figure 9.2).

In the mesh adaptation method of this chapter, the detection of vortices is translated in a field of multiplicative factors of the local mesh size. This field is depicted in Figure 9.2.

	Mesh	$\overline{C}_d$	C_l'	-$\overline{C}_{pb}$	St
Present adaptive simulation					
NF LESWale	**adap**	**0.99**	**0.045**	**0.78**	**0.20**
adaptive simulation (Chap 6)					
NF VmsWale	**adap**	**1.02**	**0.09**	**0.85**	**0.20**
Other simulations					
LES [104, 1, 68, 108]		[.99-1.38]		[.89-1.23]	[.19-0.21]
Experiments					
Ong [69]					.210 ± .05
Parnaudeau [104]					.208 ± .002
Lourenço [86]					
Cardell [30]					.215
Norberg from [1]		.99 ± .05	.09	.88 ± .005	
Kravchenko[1]		.99 ± .05		.87 ± .05	.21 ± .1

Table 9.1 – Bulk coefficient of the flow around a circular cylinder at Reynolds number **3900**. $\overline{C}_d$ holds for the mean drag coefficient, C_l' is the root mean square of lift time fluctuation, $\overline{C}_{p_b}$ is the mean pressure coefficient at cylinder basis.

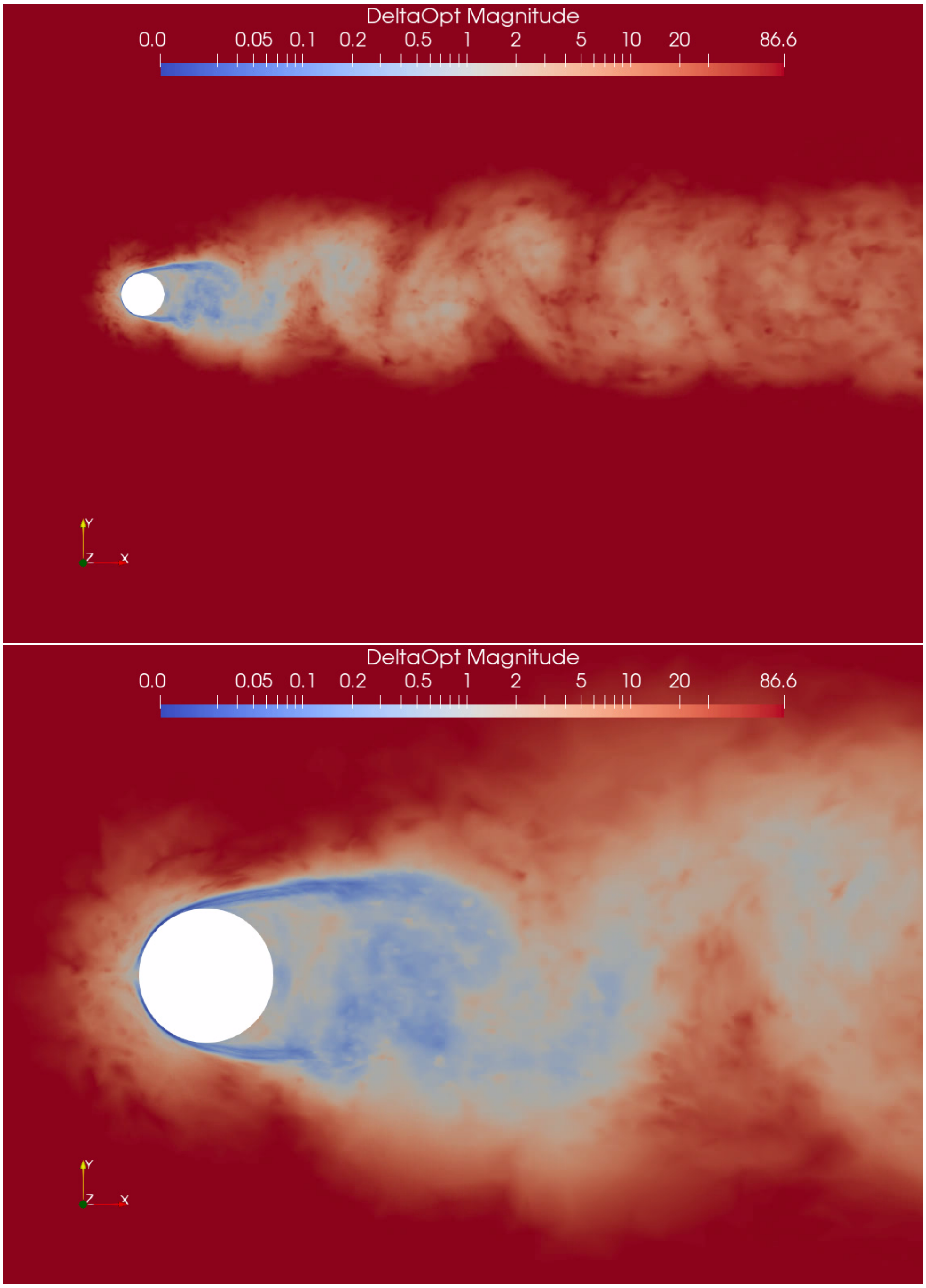


FIGURE 9.1 – *Instantaneous value of* $\Delta_1^{opt}/\bar{\Delta}$ *for flow around a cylinder at* $Re = 3900$. *These values are those given by the algorithm without clamping.*

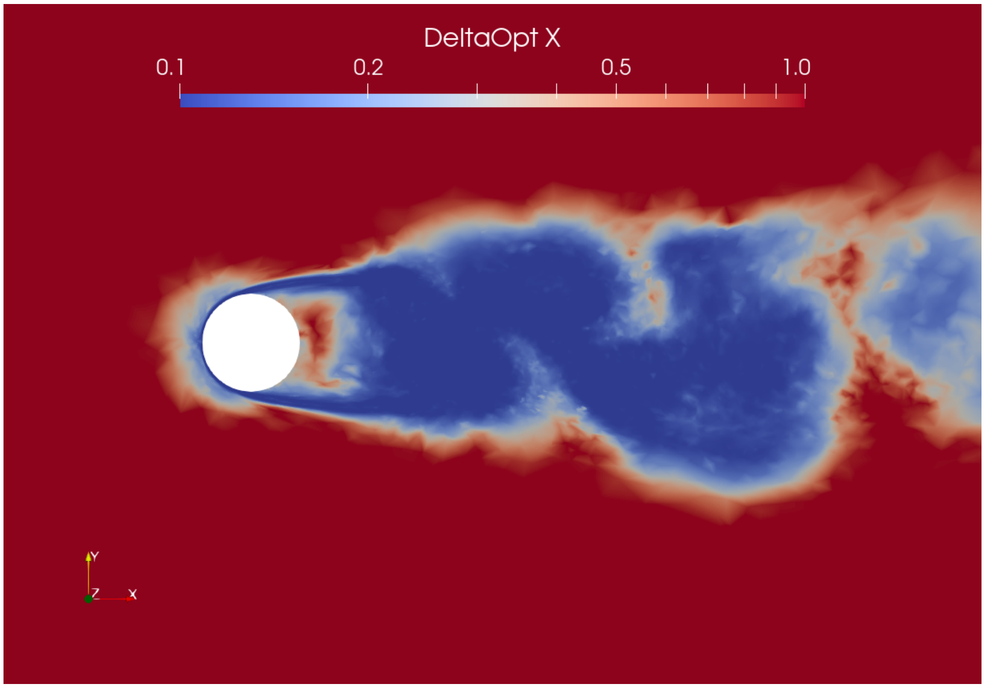


FIGURE 9.2 – *Instantaneous value of* $\Delta_1^{opt}/\bar{\Delta}$ *for flow around a cylinder at* $Re = 3900$. *These values are those given by the algorithm with a choosen clamping.*

The resulting mesh is presented in Figure 9.3 where the region one diameter after the cylinder received a supplement of vertices.

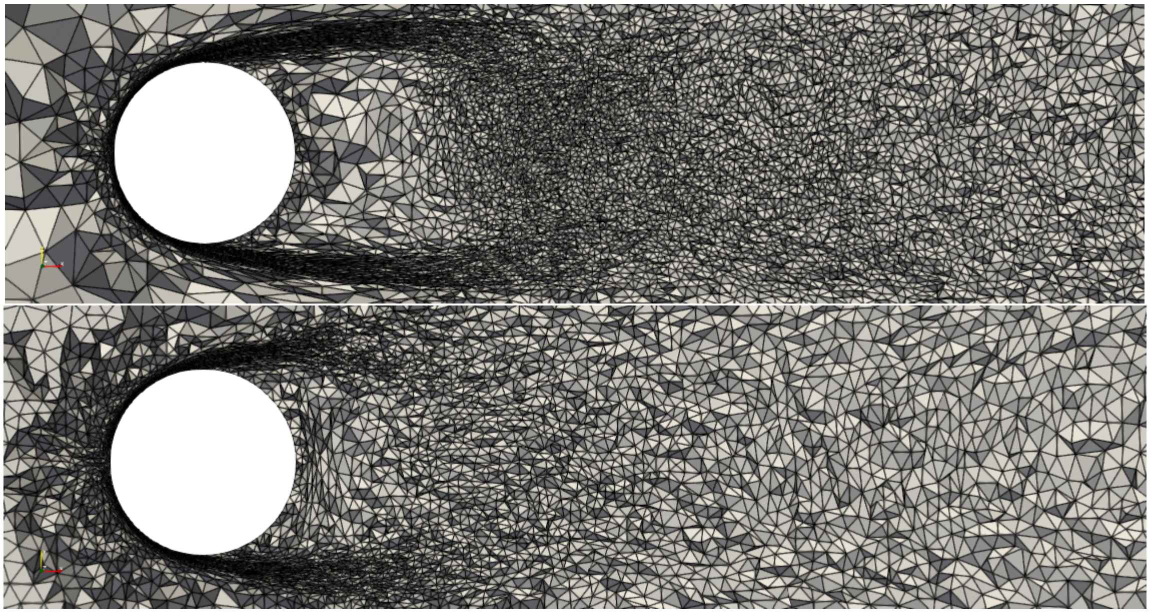


FIGURE 9.3 – *Cylindre Re = 3900 VMS-WALE results : Top, mesh obtained with the new LES criterion (around 400K vertices), and bottom, mesh obtained with Mach criterion only (around 300K vertices).*

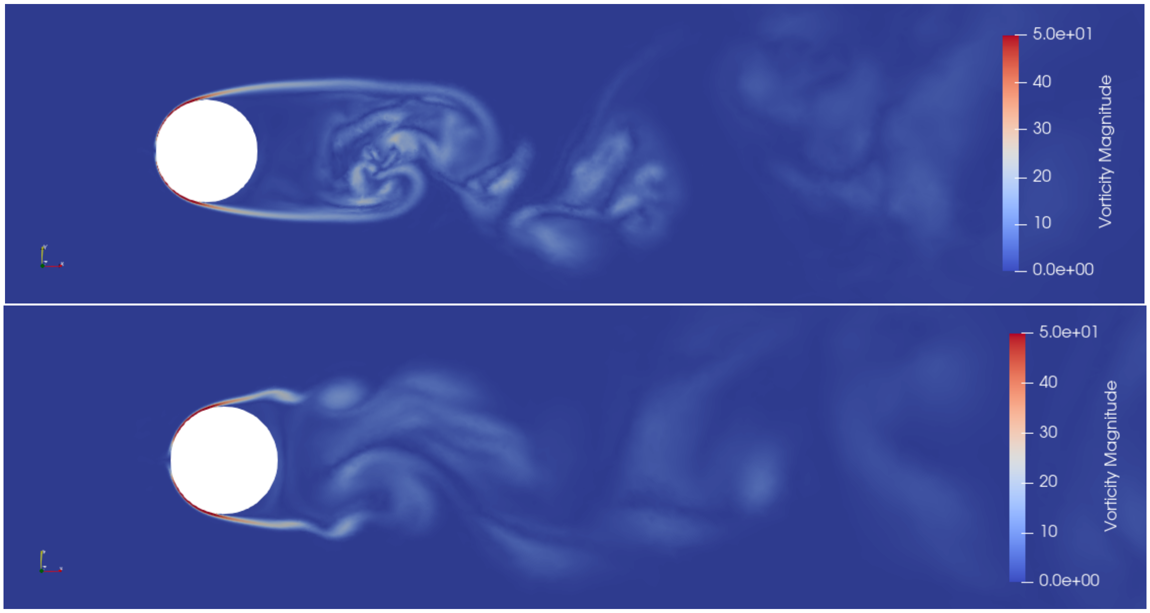


FIGURE 9.4 – *Cylindre Re = 3900 VMS-WALE results : Top, vorticity obtained with the new LES criterion, and bottom, vorticity obtained with Mach criterion only.*

6.1.2 Re = 20 000

For this test case, we compare the results obtained in Chapter 6 with those obtained for our new method. Looking at the meshes obtained in Figure 9.6, we do not see any drastic change, even if the back body for the mesh obtained with our new theory seems more remeshed. Nevertheless, Figure 1 shows that our vorticity field is captured with much greater precision.

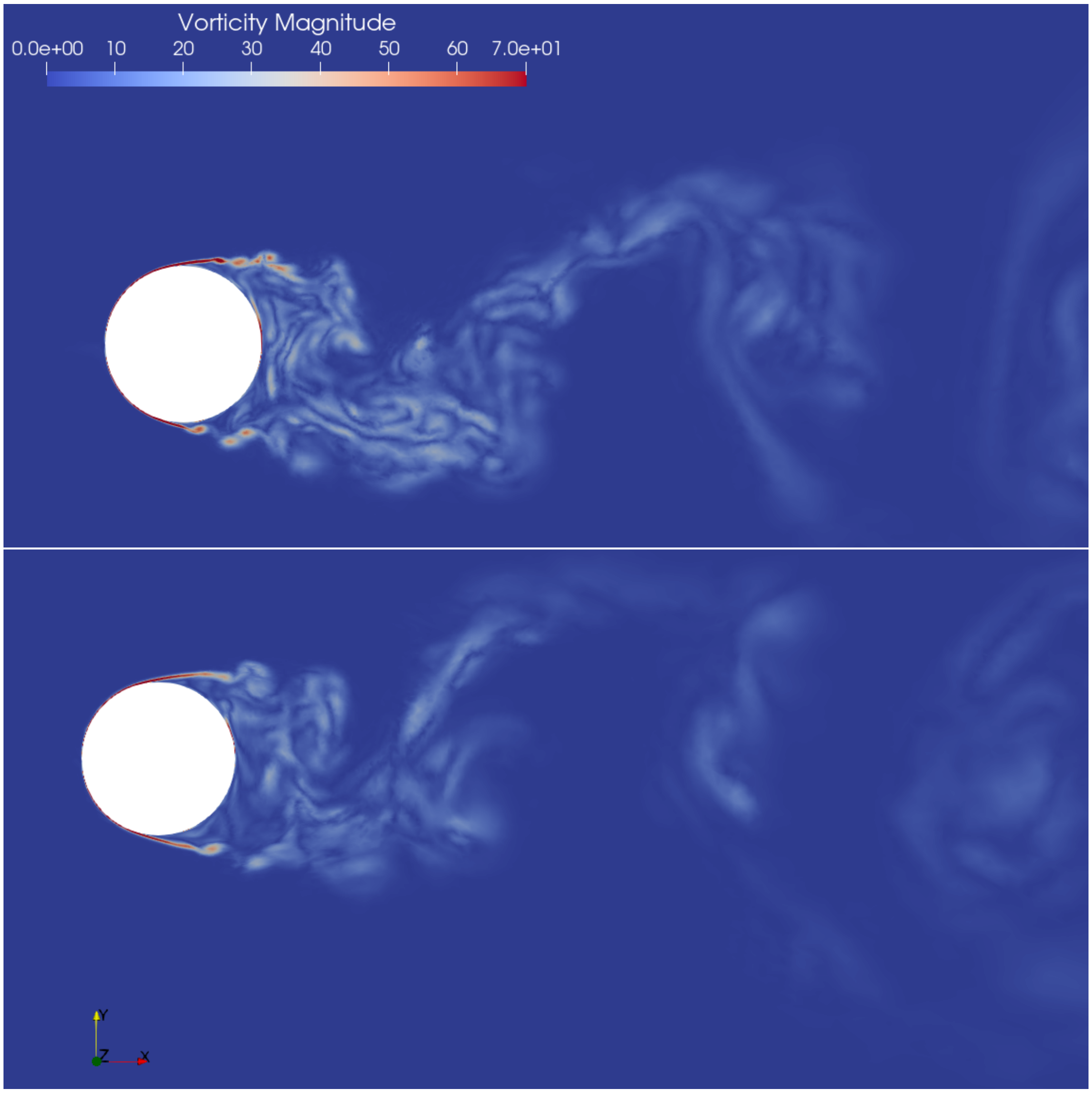


FIGURE 9.5 – *Cylindre Re = 20K VMS-WALE results : Top, vorticity obtained with the new LES criterion, and bottom, vorticity obtained with Mach criterion only.*

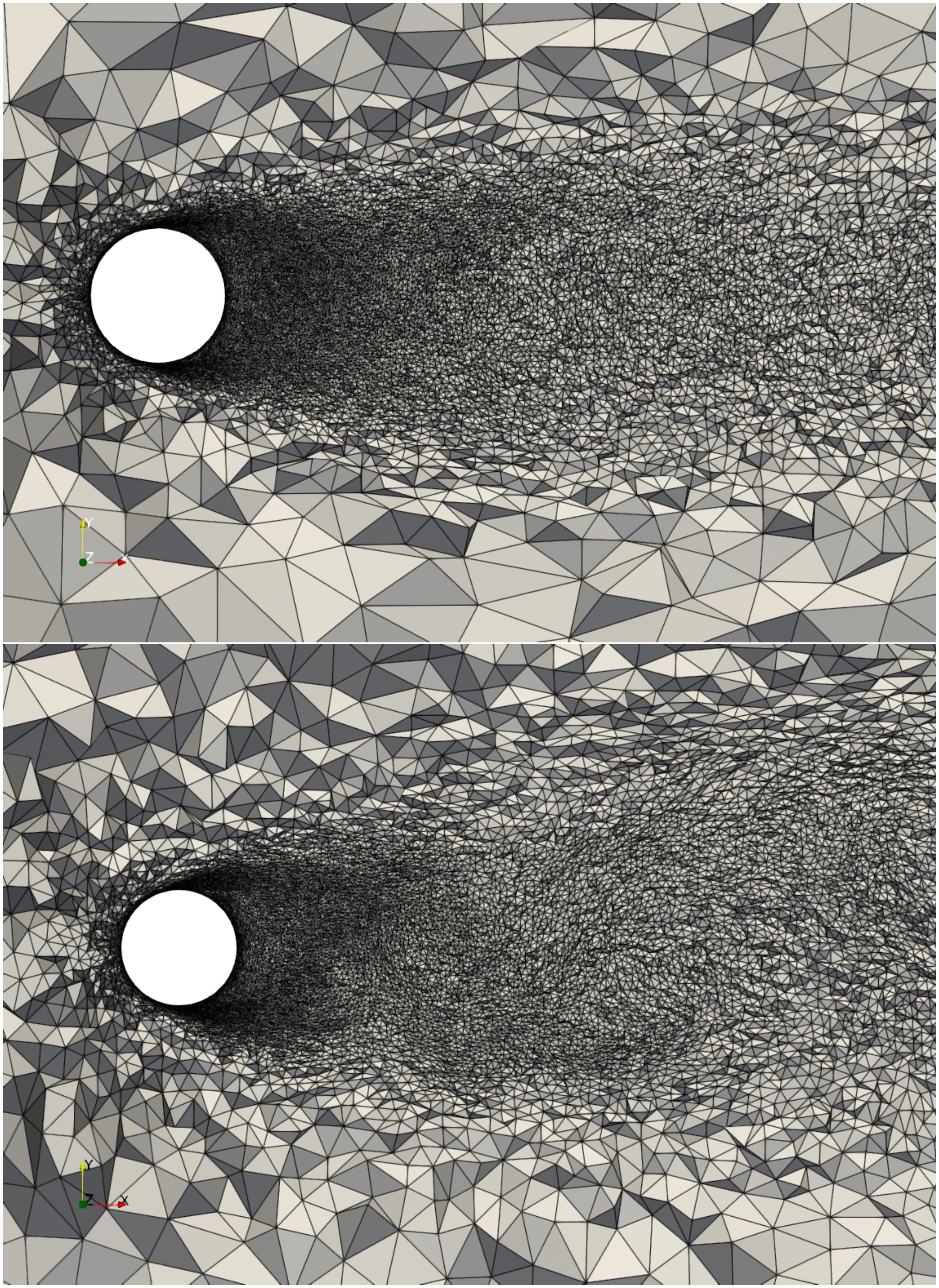


Figure 9.6 – *Cylindre Re = 20K VMS-WALE results : Top, mesh obtained with the new LES criterion (around 1M vertices), and bottom, mesh obtained with Mach criterion only (around 1.1M vertices).*

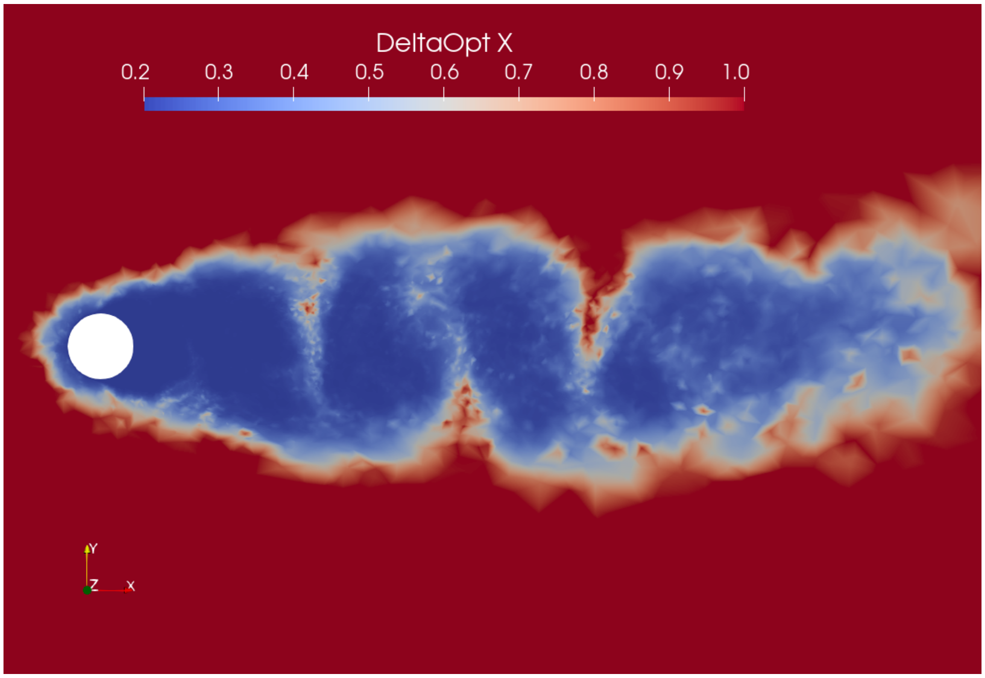


FIGURE 9.7 – *Instantaneous value of* $\Delta_1^{opt}/\bar{\Delta}$ *for flow around a cylinder at* $Re = 20K$*. These values are those given by the algorithm with a choosen clamping.*

	Mesh	$\overline{C}_d$	C_l'	-$\overline{C}_{pb}$	St
Present simulation adaptive					
NF LESWale	**adap**	**1.33**	**0.63**	**1.22**	**0.194**
simulation adaptive (Chap 6)					
NF VmsWale	**adap**	**1.24**	**0.45**	**1.08**	**0.193**
Other simul.					
LES Aradag[15]				$[1.04 - 1.25]$	
LES Salvatici[115]		$[0.94 - 1.28]$	$[0.17 - 1.65]$	$[0.83 - 1.38]$	
AN VmsWale [63]		1.17	.42	1.20	.20
Experiments					
Norberg [100]			0.47		0.194
Lim [79]		1.2		1.08	
Yokuda [139]				1.	
Basu [21]		1.17			

TABLE 9.2 – Bulk coefficients of the flow around a circular cylinder at Reynolds number **20**K. $\overline{C}_d$ holds for the mean drag coefficient, C_l' is the root mean square of lift time fluctuation, $\overline{C}_{p_b}$ is the mean pressure coefficient at cylinder basis.

6.2 Mesh adaptive computation of a supercritical flow

At a Reynolds of 1M, the flow around the circular cylinder is supercritical, with a completely turbulent boundary layer. Computing with LES this boumdary layer is very expansive, and therefore such a strategy is not useful in industrial studies. However, this type of flow is frequently met in industry.

As options in NiceFlow, we start from the anisotropic mesh adaptation approach which was proven as very efficient for steady RANS flows, see for example [8],[39], and unsteady RANS, see [13].

For addressing both RANS regions and LES-type regions, we apply the DDES available in NiceFlow, defined initially in [127] also described in Chapter 3. This model is hybrid in the sense that in some "LES" regions, the model viscosity is very small, comparable to a SGS viscosity, while not being a Smagorinsky-like viscosity.

The rest of the algorithm which we apply is identical to the one applied to the two previous subcritical flows, with two stages of adaptation criteria : a Mach-Hessian criterion and its correction with the Toosi-Larsson analysis.

At this Reynolds number and with a limited number of vertices (around 1M), the DDES calculation produces a poorly fluctuating flow, although the model DDES viscosity is much lower in the wake than the RANS viscosity. We estimate that a more fluctuating solution would be obtained by running DDES with something like 3 times mode vertices. Our development was done on a lap top with 4 processors. A more ambitious calculation is starting but will take weeks.

However, the results are reasonable without the LES criterion and improved with the LES criterion.

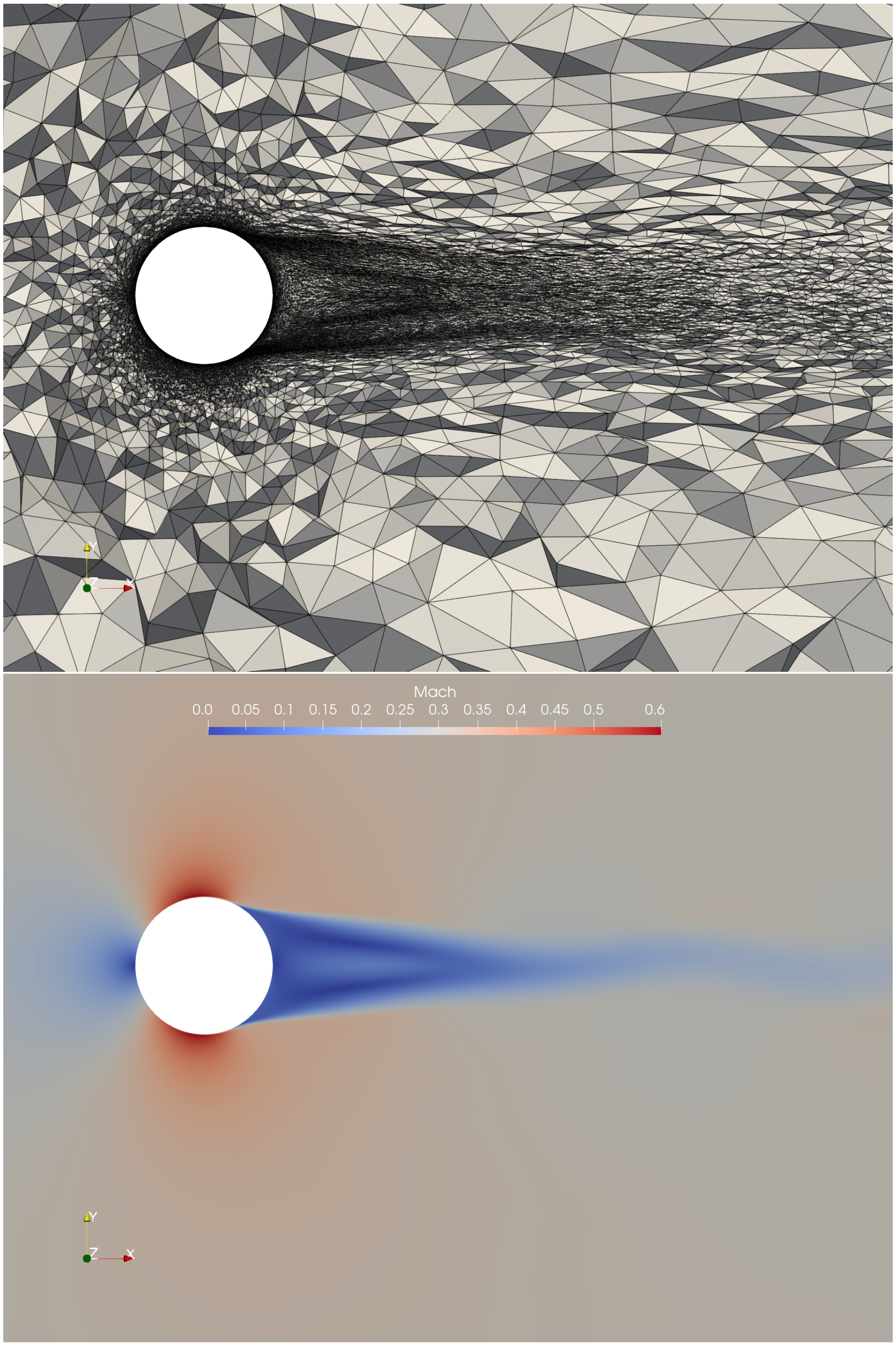


FIGURE 9.8 – *Cylindre Re = 1M DDES results : Mesh (left) and Mach field (right) in cross-section for the new LES criterion. The mesh is around 1M vertices.*

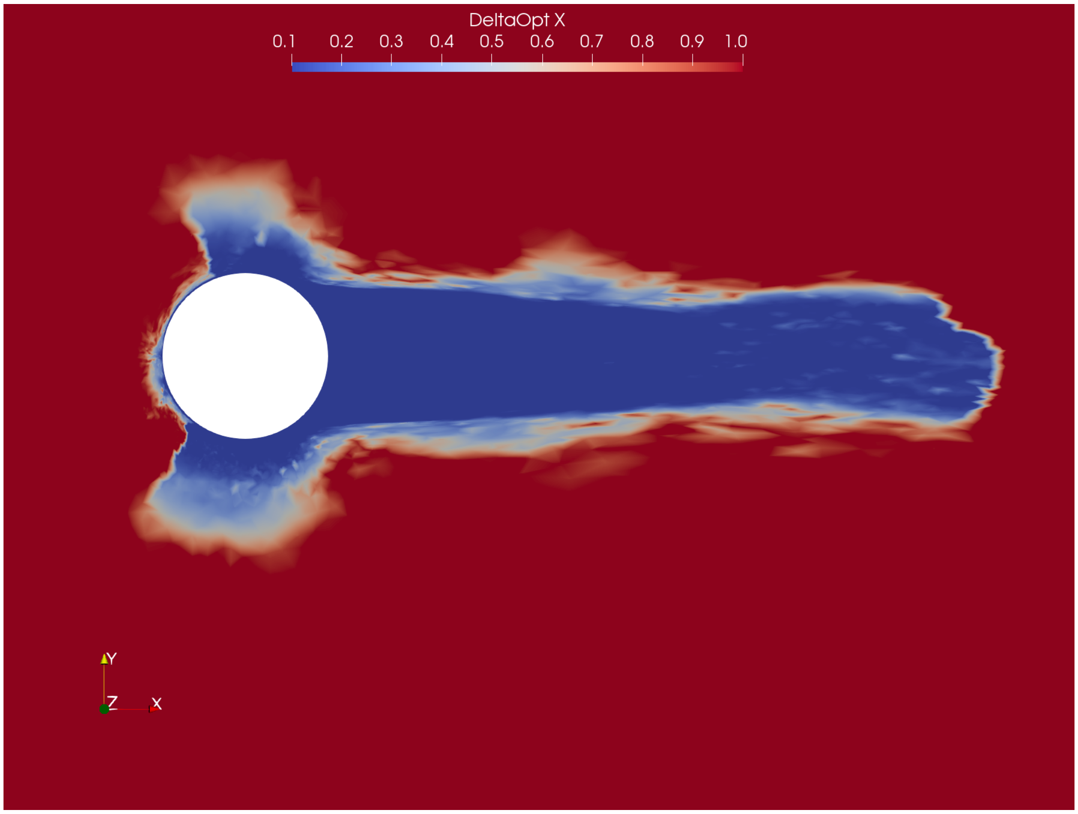


FIGURE 9.9 – *Instantaneous value of* $\Delta_1^{opt}/\bar{\Delta}$ *for flow around a cylinder at* $Re = 1M$. *These values are those given by the algorithm with a choosen clamping.*

7 Concluding remarks

Anisotropic mesh adaptive CFD is an arising important progress for applied numerical CFD. Our contribution aims at making more available hybrid RANS/LES calculations for industry, thanks to anisotropic mesh adaptation.

To address the hybrid modelization, we propose a hybrid mesh adaptation. The RANS mesh adaptation contributes to a good capture of the overall flow. For LES regions, we adapt and extend the analysis of Toosi and Larsson which is based on the source term of LES-error equation. The two mesh-adaptation criteria are applied inside a transient fixed point algorithm with the option of using a single adapted mesh for the whole time interval. Numerical examples concerning subcritical and supercritical vortex shedding flows past a circular cylinder demonstrate the interest of the proposed approach.

An extremely interesting sequel of this developement would be to extend the analysis to the so-called goal-oriented analysis. In [28], the goal oriented analysis permitted to look for the best mesh for propagating an acoustic wave produced by an artificial acoustic source. With the new criterion, noise production by turbulence could be addressed.

The introduction of the space-time adaptation would be also of interest in order to compute with optimally chosen time steps.

CHAPITRE 10

Conclusion

The objectives of this thesis were multiple to meet certain expectations of the Norma project. Indeed, we discussed the subject of rotating machines, mesh adaptation problems for unsteady turbulent flows, but also performance problems that arise in unsteady mesh adaptation... Let us see what we have achieved.

The Chapter 6 have permitted to elaborate anisotropic mesh adaptive strategies for a first families of turbulent LES/hybrid flows. The efficiency and accuracy of the steady RANS mesh adaptive case was already well established in a series of papers and in the book [39]. For LES and hybrid simulations of vortex shedding flows like those considered in this Chapter 6, the work in this thesis has identified a first mesh adaptation strategy. We keep the feature-based Mach-number criterion. The flow is now an unsteady one. We use the Transient fixed point mesh adaptation algorithm, but with only the mesh for the whole time interval (it is changed by the fixed point). Preliminary results show that this approach is of reasonable efficiency.

In Chapter 7 we looked at different methods for taking rotation into account in our simulations, and settled on the Multiple Reference Frame method for reasons of performance and consideration of our applications. This method has been successfully applied to a test case of a helicopter rotor coupled with mesh adaptation. We obtained good results for both the stationary flow and the unsteady DDES test. However, we didn't have the time to study rotor/fuselage interaction cases in depth. A simple computational example is given for our Caradonna-Tung rotor with a Robin fuselage, but there are no experimental results for this hovering case, only with a non-zero flight speed. Naturally, the continuation of the work in this section, within the framework of the Norma project, would be the acoustic study of our test case using our mesh adaptation method. But above all, to extend our studies to rotor/fuselage cases with one or more rotors, and also in non-stationary flight.

In Chapter 8 we present our new theory of space-time mesh adaptation. In this work, we have presented a Global Space-Time Transient Fixed Point (GSTTFP) method which automatically provides the optimal space-time adapted mesh, i.e., it provides the optimal spatial adapted mesh and the optimal time steps, to minimize the considered space-time error model. Two error models have been theoretically considered : the feature-based error model based on a control or the space-time interpolation error in Lp norm of a given sensor and the goal-oriented error model based on the control of the space-time approximation error of a given scalar output functional. This work extends the previous Global Transient Fixed Point methods (GTFP). It should be noted that the new GSTTFP method can be directly used in place of the existing GTFP approach resulting in a direct

CPU/accuracy improvement induced by the optimal choice of time step. The proposed method, in the feature error model context, has been validated on 2D and 3D simulations of turbulent flow past a cylinder at different Reynolds number regimes. First, we have pointed out the fast iterative convergence of the GSTTFP algorithm as just a few fixed point iterations are required to obtain the optimal spatial and temporal adapted meshes. In all cases, the spatial and temporal errors have been almost balanced by the algorithm. The gap between both error components decreases when the space-time complexity increases. We have also observed that the temporal adapted mesh captures the dynamic of the flow, in particular, the periodicity in time of the flow is clearly visible in the temporal adapted mesh. As regards the accuracy of the method, a space-time convergence order of ~ 1.8 has been observed for the 2D simulations using $n_{adap} = 1$. However, the $n_{adap} \geqslant 1$ case has not been studied to any great extent, due to technical problems that have yet to be resolved. The perspectives of this work are numerous. First, the case where n_{adap} higher than unity requires a detailed numerical study for other types of applications involving, for instance, rapid dynamics, although some results have been presented in this paper. In that context, the variable n_{adap} changes during the mesh convergence study (i.e., when the space-time complexity increases) in order to optimize the space-time mesh and increase the order of convergence according to the theoretical study described for the Global Transient Fixed Point methods. Second, the proposed error optimization is presented in a general formulation applying to either feature-based criteria or goal-oriented criteria. The important goal-oriented case needs further study and development as it requires the solution of the backward in time adjoint problem. This will be experimented in a future work. Lastly, the proposed approach for quasi-steady flow using only one adapted mesh for the entire simulation time frame is very suitable for Large Eddy Simulation (LES) applications. The extension of this work to LES is therefore natural and will be considered in the future.

In Chapter 9, in order to address the hybrid modelization, we propose a hybrid mesh adaptation. The RANS mesh adaptation contributes to a good capture of the overall flow. For LES regions, we adapt and extend the analysis of Toosi and Larsson which is based on the source term of LES-error equation. The two mesh-adaptation criteria are applied inside a transient fixed point algorithm with the option of using a single adapted mesh for the whole time interval. Numerical examples concerning vortex shedding flows past a circular cylinder at Reynolds numbers 3900 and 20K demonstrate the interest of the proposed approach. An extremely interesting sequel of this developement would be to extend the numerical validation of our theory to the so-called goal-oriented analysis. The goal oriented analysis permitted to look for the best mesh for propagating an acoustic wave produced by an artificial acoustic source. With the new criterion, noise production by turbulence could be addressed.

A natural future extension of all this work is, of course, to bring these different methods together. It would then be possible to carry out rotor/stator type simulations with greater precision, saving CPU and human time, and with better capture of LES/vortex phenomena.

Bibliographie

[1] P. Moin A. Kravchenko. Numerical studies of flow over a circular cylinder at Re = 3900. Physics of Fluids, 12 :2 :403-17, 2000.

[2] M. Strelets P. R. Spalart A. Travin, M. L. Shur. Physical and numerical upgrades in the detached-eddy simulation of complex turbulent flows. In :412 EUROMECH Colloquium on LES of complex transitional and turbulent flows Munich, 2000.

[3] I. Abalakin, V. Bobkov, V.Tsvetkova, B. Sauvage, F. Miralles, T. Kozubskaya, S.F. Wornom, B. Koobus, and A. Dervieux. Towards efficient simulation of turbulent flows and noise in rotating machines. 22nd Computational Fluids Conference ; CFC 2023, 2023.

[4] R. Abgrall. Design of an essentially non-oscillatory reconstruction procedure on Finite-Element type meshes. Rapport technique 1584, INRIA, 1992.

[5] R. Abgrall. Residual distribution schemes : current status and future trends. Computers and Fluids, 35 :641–669, 2006.

[6] E. Achenbach. Total and local heat transfer from a smooth circular cylinder in cross-flow at high reynolds number. International Journal of Heat and Mass Transfer, 18 :1387–1396, 1975.

[7] R. El Akoury, M. Braza, C. Mockett, R. Perrin, T. Reimann, A. Revell, T. Craft, D. Laurence, Y. Hoarau, and F. Thiele. Iv applications - test cases, 1 circular cylinder flow. Notes on Numerical Fluid Mechanics and Multidisciplinary Design, 103, 2009.

[8] F. Alauzet and L. Frazza. 3D RANS anisotropic mesh adaptation on the high-lift version of NASA common research model (HL-CRM). AIAA paper 2019-2947, 2020.

[9] F. Alauzet and L. Frazza. Feature-based and goal-oriented anisotropic mesh adaptation for RANS applications in aeronautics and aerospace. Journal of Computatiuonal Physics, 439 :110340, 2021.

[10] 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. of Computational Physics, 222 :592–623, 2007.

[11] F. Alauzet, P.J. Frey, and B. Mohammadi. Adaptation de maillages non structurés pour des problèmes instationnaires. C.R. Acad. Sci. ParisSer. I, 303 :773–778, 2002. in French.

[12] 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, 43(6-7) :729–745, 2003.

[13] F. Alauzet, A. Loseille, and G. Olivier. Time-accurate multi-scale anisotropic mesh adaptation for unsteady flows in CFD. Journal of Computational Physics, 373 :28–63, 2018.

[14] J. Andren, H. Gao, M. Yano, D.L. Darmofal, C. Ollivier-Gooch, and Z.J. Wang. A comparison of higher-order methods on a set of canonical aerodynamics applications. 20th AIAA Computational Fluid Dynamics Conference 27 - 30 June 2011, Honolulu, Hawaii, AIAA 2011-3230 :1–13, 2011.

[15] S. Aradag. Unsteady turbulent vortex structure downstream of a three dimensional cylinder. J. of Thermal Science and Technology, 29(1) :91–98, 2009.

[16] P. Bakhvalov and T. Kozubskaya. EBR-WENO scheme for solving gas dynamics problems with discontinuities on unstructured meshes. Computers and Fluids, 157(3) :312–324, 2017.

[17] T.J. Barth. Recent developments of high-order k-exact reconstruction on unstructured meshes. In 31st AIAA Aerospace Science Meeting, AIAA-93-0668, Reno, NV, USA, 1993.

[18] T.J. Barth and P.O. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA-90-0013, 1990.

[19] F. Basile, J.-B. Chapelier, R. Laraufie, and P. Frey. hp-adaptive hybrid RANS/LES simulations for unstructured meshes with the discontinuous galerkin method. AIAA SCITECH 2022 Forum, Jan 2022, San Diego, United States. 10.2514/6.2022-1207. hal-03632119, 2022.

[20] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. Journal of Computational Physics, 138(2) :251–285, 1997.

[21] R.I. Basu. Aerodynamic forces on structures of circular cross-section. part i : Model-scale data obtained under two-dimensional and low-turbulence streams. J. Wind Eng. Ind. Aerodynamic, 21 :273–294, 1985.

[22] A. Belme, F. Alauzet, and A. Dervieux. A priori anisotropic goal-oriented estimate and mesh adaptation for viscous compressible flow. Preprint, 2017.

[23] A. Belme, A. Dervieux, and F. Alauzet. Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows. J. Comp. Phys., 231(19) :6323–6348, 2012.

[24] S. Bharara. Simulation of drag crisis in flow past a circular cylinder using 3-d computations. International Journal of Aerospace and Mechanical Engineering, 3 :17–19, 2016.

[25] M. Blind, A. Berk Kahraman, J. Larsson, and A. Beck. Grid-adaptation for wall-modeled Large Eddy Simulation using unstructured high-order methods. arXiv :2301.03199v1, 2023.

[26] S. Camarri, B. Koobus, M.-V. Salvetti, and A. Dervieux. A low-diffusion MUSCL scheme for les on unstructured grids. Computers and Fluids, 33 :1101–1129, 2004.

[27] 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, 40 :1431–1460, 2002.

[28] A. Carabias, A. Belme, A. Loseille, and A. Dervieux. Anisotropic goal-oriented error analysis for a third-order accurate CENO Euler discretization. International Journal for Numerical Methods in Fluids, 86(6) :392–413, 2018.

[29] F. X. Caradonna and C. Tung. Experimental and analytical studies of a model helicopter rotor in hover. Technical Report NASA-TM-81232, NASA, Ames Research Center, Moffett Field, California, September 1981.

[30] G.S. Cardell. Flow past a circular cylinder with a permeable splitter plate. PhD thesis, Graduate Aeronautical Lab., California Inst. of Technology, 1993.

[31] B. Chaouat. The state of the art of hybrid RANS/LES modeling for the simulation of turbulent flows. Flow Turbulence Combust., 99 :279–327, 2017.

[32] M.R.J. Charest, C.P.T. Groth, and P.Q. Gauthier. A high-order central ENO finite-volume scheme for three-dimensional low-speed viscous flows on unstructured meshs. Communications in Computational Physics, 17(03) :615–656, 2015.

[33] P.G. Ciarlet and P.A. Raviart. General Lagrange and Hermite interpolation in R^n with applications to Finite Element methods. Archive for Rational Mechanics and Analysis, 46 :177–199, 1972.

[34] B. Cockburn. Devising Discontinuous Galerkin methods for non-linear hyperbolic conservation laws. Journal of Computational and Applied Mathematics, 128(1-2) :187–204, 2001.

[35] B. Cockburn, G. Karniadakis, and C.-W. Shu. Discontinuous Galerkin methods : theory, computation and application. Lecture Notes in Computational Science and Engineering. Springer Verlag, Berlin, 2000.

[36] T. Coupez, G. Jannoun, N. Nassif, H.C. Nguyen, H. Digonnet, and E. Hachem. Adaptive time-step with anisotropic meshing for incompressible flows. J. Comp. Phys., 241 :195–211, 2013.

[37] J. Kampe de Feriet. Average processes and Reynolds equations in atmospheric turbulence. Journal of the Atmospheric Sciences, Volume 8 : Issue 5 :358–361, 1951.

[38] C. Debiez and A. Dervieux. Mixed element volume muscl methods with weak viscosity for steady and unsteady flow calculation. Computer and Fluids, 29 :89–118, 1999.

[39] A. Dervieux, F. Alauzet, A. Loseille, and B. Koobus. Mesh adaptation for Computational Fluid Dynamics, t.1 and t.2. ISTE Ltd and John Wiley & Sons, 1st edition, 2023.

[40] L. Dubuc, F. Cantariti, M. Woodgate, B. Gribben, K.J. Badcock, and B.E. Richards. A grid deformation technique for unsteady flow computations. Int. J. Num. Meth. Fluids, 32 :285–311, 2000.

[41] S. Engquist, B. Harten, A. Osher, and S.R. Chakravarthy. Some results on uniformly high-order accurate essentially non oscillatory schemes. Appl. Numer. Math. 2(3-5) :347-377, 1986.

[42] G. Erlebacher, M. Y. Hussaini, C. G. Speziale, and T. A. Zang. Toward the large-eddy simulation of compressible turbulent flows. Journal of Fluid Mechanics, 238 :155 – 185, 1992.

[43] A. Harten et S. Chakravarthy. Multi-dimensional ENO schemes for general geometries. ICASE report 91-76, 1991.

[44] C. Farhat. High performance simulation of coupled nonlinear transient aeroelastic problems. Special course on parallel computing in CFD R-807, NATO AGARD Report, October 1995.

[45] C. Farhat, P. Geuzaine, and C. Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. J. of Computational Physics, 174 :669–694, 2001.

[46] K. J. Fidkowski. Output-based error estimation and mesh adaptation for unsteady turbulent flow simulations. Computer Methods in Applied Mechanics and Engineering, 2022.

[47] C. E. Freeman and R. E. Mineck. Fuselage surface pressure measurements of a helicopter wind-tunnel model with a 3.15-meter diameter single rotor. Technical Report NASA-TM-80051, NASA, NASA Langley Research Center Hampton, VA, 1979.

[48] J. Froehlich and D. von Terzi. Hybrid LES/RANS methods for the simulation of turbulent flows. Prog. Aerosp. Sci., 44(5) :349–377, 2008.

[49] F. Gatti, M. Fois, C. de Falco, S. Perotto, and L. Formaggia. Parallel simulations for fast-moving landslides : space-time mesh adaptation and sharp tracking of the wetting front. International Journal for Numerical Methods in Fluids, 95 :1286–1309, 2023.

[50] P.L. George, F. Hecht, and M.G. Vallet. Creation of internal points in Voronoi's type method. Control and adaptation. Adv. Eng. Software, 13(5-6) :303–312, 1991.

[51] M. Germano, U. Piomelli, P. Moin, and W. Cabot. A dynamic subgrid-scale eddy viscosity model. Phys. Fluids A, 3(7) :1760–1765, 1991.

[52] Sandip Ghosal. On the effect of numerical errors in large eddy simulations of turbulent flows. Journal of Computational Physics, 131 :310–322, 1997.

[53] U. Goldberg, O. Peroomian, and S. Chakravarthy. A wall-distance-free $k-\varepsilon$ model with enhanced near-wall treatment. Journal of Fluids Engineering, 120 :457–462, 1998.

[54] B. Gölling. Experimental investigations of separating boundary-layer flow from circular cylinder at reynolds numbers from 10^5 up to 10^7 ; three-dimensional

vortex flow of a circular cylinder. In G.E.A. Meier and K.R. Sreenivasan, editors, Proceedings of IUTAM Symposium on One Hundred Years of Bloundary Layer Research, pages 455–462, The Netherlands, 2006. Springer.

[55] C.P.T. Groth and L. Ivan. High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition AIAA 2011-367 4 - 7 January 2011, Orlando, Florida, 2011.

[56] M. Gschwend. A 3D vertex centered CENO scheme for advection. RR-9415, Inria Sophia Antipolis-Méditerranée, hal-03284753, 2021.

[57] 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. Engng, 84(11) :1376–1406, 2010.

[58] A. Dervieux MV. Salvetti H. Ouvrard, B. Koobus. Classical and variational multiscale les of the flow around a circular cylinder on unstructured grids. Comput Fluids, 39 :1083-94, 2010.

[59] A. Hauser and G. Wittum. Adaptive Large Eddy Simulation. Comput Visual Sei, 17 :295-304 DOI 10.1007/s00791-016-0265-3, 2015.

[60] T.J.R. Hughes, G.R. Feij, L. Mazzei, and J.-B. Quincy. The variational multiscale method-a paradigm for computational mechanics. Comput. Methods Appl. Mech. Engrg, 166 :3–24, 1998.

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

[62] Jeremy Ims and Z.J. Wang. A comparison of three error indicators for adaptive high-order Large Eddy Simulation. Journal of Computational Physics, 490 :112312, 2023.

[63] E. Itam. Simulation numérique découlements autour de corps non profilés par des modèles de turbulence hybrides et un schéma multirate. PhD thesis, Université Montpellier II, Montpellier, France, 2017. (in French).

[64] L. Ivan and C.P.T. Groth. High-order central eno finite-volume scheme with adaptive mesh refinement. 18th AIAA Computational Fluid Dynamics Conference 25 - 28 June 2007, Miami, FL, (AIAA 2007-4323), 2007.

[65] L. Ivan and C.P.T. Groth. High-order central ENO finite-volume scheme with adaptive mesh refinement. In 18th AIAA Computational Fluid Dynamics Conference, AIAA-2007-4323, 25 - 28 June 2007, Miami, FL, 2007.

[66] L. Ivan and C.P.T. Groth. High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. J. Comp. Phys., 257 :830–862, 2014.

[67] L. Ivan, H. De Sterck, and C.P.T. Groth. A fourth-order solution-adaptive CENO scheme for space-physics flows on three-dimensional multi-block cubed-sphere grids. Conference : 22nd Annual Conference of the CFD Society of Canada, 2014.

[68] S. Lee H. Choi J. Lee, N. Park. A dynamical subgrid-scale eddy viscosity model with a global model coefficient. Physics of Fluids, 18 :12, 2006.

[69] L. Wallace J. Ong. The velocity field of the very near wake of a circular cylinder. Exp Fluids, 20 :441, 1996.

[70] B. Khara, K. Saurabh, R. Dyja, A. Sharma, and B. Ganapathysubramanian. Space-time finite element analysis of the advection-diffusion equation using Galerkin/least-square stabilization. arXiv :2307.00822 [math.NA], https ://arxiv.org/pdf/2307.00822.pdf, 2023.

[71] S.-E. Kim and L.S. Mohan. Prediction of unsteady loading on a circular cylinder in high reynolds number flows using large eddy simulation. Proceedings of OMAE 2005 : 24th International Conference on Offshore Mechanics and Artic Engineering, june 12-16, Halkidiki, Greece, OMAE 2005-67044, 2005.

[72] James L Kocurek, J. David ; Tangler. A prescribed wake lifting surface hover performance analysis. Journal of the American Helicopter Society, Volume 22, Number 1 :24–35(12), 1977.

[73] 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. Computer Methods in Applied Mechanics and Engineering, 193 :1367–1383, 2004.

[74] A. G. Kravchenko and P. Moin. An analysis of numerical errors in Large-Eddy Simulations of turbulence. Journal of Computational Physics, 125 :187–206, 1996.

[75] F.C. Lafon and R. Abgrall. ENO schemes on unstructured meshes. Rapport INRIA 2099, 1993.

[76] O. Lehmkuhl, I. Rodriguez, R. Borrell, J. Chiva, and A. Oliva. Unsteady forces on a circular cylinder at critical reynolds numbers. Physics of Fluids, 26 :125110, 2014.

[77] M. Lesieur and O. Métais. New trends in large-eddy simulations of turbulence. Annu. Rev. Fluid Mech., 28 : :45–82, 1996.

[78] D.K. Lilly. A proposed modification of the germano subgrid-scale closure method. Phys. Fluids, A4 :633, 1992.

[79] H. Lim and S. Lee. Flow control of circular cylinders with longitudinal grooved surfaces. AIAA Journal, 40(10) :2027–2035, 2002.

[80] T.P. Lloyd and M. James. Large eddy simulations of a circular cylinder at reynolds numbers surrounding the drag crisis. Applied Ocean Research, 59 :676–686, 2015.

[81] R. Löhner and J.D. Baum. Adaptive h-refinement on 3D unstructured grids for transient problems. Int. J. Numer. Meth. Fluids, 14 :1407–1419, 1992.

[82] A. Loseille and F. Alauzet. Continuous mesh framework. Part I : well-posed continuous interpolation error. SIAM Num. Anal., 49(1) :38–60, 2011.

[83] A. Loseille and F. Alauzet. Continuous mesh framework. Part II : validations and applications. SIAM Num. Anal., 49(1) :61–86, 2011.

[84] A. Loseille, A. Dervieux, and F. Alauzet. Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. J. Comp. Phys., 229 :2866–2897, 2010.

[85] A. Loseille and R. Löhner. Cavity-based operators for mesh adaptation. 51st AIAA Aerospace Sciences Meeting, Jan 2013.

[86] L.M. Lourenco and C. Shih. Characteristics of the plane turbulent near weak of a circular cylinder. a particle image velocimetry study, (data taken from kravchenko and moin).

[87] Y. Luo and K. J. Fidkowski. Output-based space-time mesh adaptation for unsteady aerodynamics. 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition 4 - 7 January 2011, Orlando, Florida, AIAA 2011-491, 2011.

[88] K. Mani and D. J. Mavriplis. Discrete adjoint based time-step adaptation and error reduction in unsteady flow problems. 18th AIAA Computational Fluid Dynamics Conference 25 - 28 June 2007, Miami, FL, AIAA 2007-3944, 2007.

[89] R. Martin and H. Guillard. A second-order defect correction scheme for unsteady problems. Comput. and Fluids, 25(1) :9–27, 1996.

[90] F. Menter, A. Hüppe, A. Matyushenko, and D. Kolmogorov. An overview of hybrid rans-les models developed for industrial cfd. Appl. Sci., 11 :2459, 2021.

[91] S. Micheletti and Simona Perotto. Space-time adaptation for purely diffusion problems in an anisotropic framework. International Journal of Numerical Analysis and Modeling, 7(1) :125–155, 2010.

[92] S. Micheletti and Simona Perotto. Time adaptivity and anisotropic mesh adaptation for CFD applications. In J. P. Moitinho de Almeida, P. Diéz, C. Tiago, and N. Parés, editors, VI International Conference on Adaptive Modeling and Simulation ADMOS 2013, 2013.

[93] F. Miralles and B. Koobus. Simulation of the flow past a circular cylinder from sub-critical to super-critical reynolds numbers using an intermittency-based hybrid model. Journal of Fluids and Structures, 123 :104008, 2023.

[94] C. Moussaed, S. Wornom, M.-V. Salvetti, B. Koobus, and A. Dervieux. Impact of Dynamic SubGrid-Scale modeling in Variational MultiScale Large-Eddy Simulation of bluff-body flows. Acta Mechanica, 225 :3309–3323, 2014.

[95] F. Naddei. Simulation adaptative des grandes échelles d'écoulements turbulents fondée sur une méthode Galerkine discontinue. Numerical Analysis [cs.NA]. Université Paris Saclay (COmUE), 2019. English. NNT : 2019SACLX060, 2019.

[96] A. Nejat and C. Ollivier-Gooch. Effect of discretization order on preconditioning and convergence of a high-order unstructured Newton-GMRES solver for the Euler equations. Journal of Computational Physics, 227 :2366 ?2386, 2008.

[97] A. Nejat and C.F. Ollivier-Gooch. A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. Journal of Computational Physics, 227(4) :2582–2609, 2008.

[98] F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. Flow Turbulence and Combustion, 62(3) :183–200, 1999.

[99] B. Nkonga and H. Guillard. Godunov type method on non-structured meshes for three-dimensional moving boundary problems. Comput. Methods Appl. Mech. Eng., 113(1-2) :183–204, 1994.

[100] C. Norberg. Fluctuating lift on a circular cylinder : review and new measurements. J. Fluids Struct., 17 :57–96, 2003.

[101] C. Ollivier-Gooch, A. Nejat, and K. Michalak. Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations. AIAA Journal, 47(9) :2105–2120, 2009.

[102] N. Botterilland J.S. Owen and H.P. Morvan. Investigation into numerical modeling of the drag crisis for circular cylinders. 5th European & African conference on wind engineering : Florence Italy, July 19th-23rd 2009, conference proceedings, 2009.

[103] G. Iaccarino P. Moin P. Catalano, M. Wang. Numerical simulation of the flow around a circular cylinder at high reynolds numbers. International Journal of Heat and Fluid Flow, 24 :463–469, 2003.

[104] D. Heitz E. Lamballais P. Parnaudeau, J. Carlier. Experimental and numerical studies of the flow over a circular cylinder at reynolds number 3900. Physics of Fluids, 20 085101, 2008.

[105] D. Pagnutti and C. Ollivier-Gooch. A generalized framework for high order anisotropic mesh adaptation. , 87 :670–679, 2009.

[106] D. Papadogiannis, L. Billon, and F. Alauzet. Isotropic and anisotropic mesh adaptation for RANS simulations of a nacelle under crosswind conditions. Proceedings of Global Power and Propulsion Society, GPPS Chania22, september 12-14, 2022. www.gpps.global,ISSN-Nr : 2504-4400, 2022.

[107] G. Pont, P. Brenner, P. Cinnella, B. Maugars, and J.-C. Robinet. Multiple-correction hybrid k-exact schemes for high-order compressible rans-les simulations on fully unstructured grids. Journal of Computational Physics, 350 :45–83, 2017.

[108] B.N. Rajani, A. Kandasamy, and Sekhar Majumdar. LES of flow past circular cylinder at Re = 3900. Journal of Applied Fluid Mechanics, 9(3) :1421–1435, 2016.

[109] R.D. Rausch, J.T. Batina, and H.T.Y. Yang. Spatial adaptation procedures on tetrahedral meshes for unsteady aerodynamic flow calculations. AIAA Journal, 30 :1243–1251, 1992.

[110] I. Rodriguez, O. Lehmkuhl, J. Chiva, R. Borrell, and A. Oliva. On the flow past a circular cylinder from critical to super-critical reynolds numbers : Wake topology

and vortex shedding, international journal of heat and fluid flow 55, 91–103. Physics of Fluids, 26 :125110, 2015.

[111] A. Roshko. Experiments on the flow past a circular cylinder at very high Reynolds number. J. Fluid. Mech., 10 :345–356, 1961.

[112] A. Ekmekci D. Rockwell S. Dong, GE. Karniadakis. A combined dns-piv study of the turbulent near wake. J Fluids Mech, 569 :185–207, 2006.

[113] M. V. Salvetti B. Koobus A. Dervieux S. Wornom, H. Ouvrard. Variational multiscale large-eddy simulations of the flow past a circular cylinder : Reynolds number effects. Computer & Fluids, 47 :44–50, 2011.

[114] P. Sagaut, S. Deck, and M. Terracol. Multiscale and multiresolution approaches in turbulence. Imperial College Press, London, 2006.

[115] E. Salvatici and M. V. Salvetti. Large-eddy simulations of the flow around a circular cylinder : effects of grid resolution and subgrid scale modeling. Wind and Structures, 6(6) :419–436, 2003.

[116] B. Sauvage, F. Alauzet, and A. Dervieux. A space and time fixed point adaptation method. Deliverable T4-D3, ANR Norma, 2022.

[117] J.W. Schewe. On the forces acting on a circular cylinder in cross flow from subcritical up to transcritical Reynolds numbers. J. Fluid Mech., 133 :265–285, 1983.

[118] F. Setzwein, P. Ess, and P. Gerlinger. High-order k-exact finite volume scheme for vertex-centered unstructured grids. AIAA 2020-1785, 2020.

[119] M. Sharbatdar and C. Ollivier-Gooch. Anisotropic mesh adaptation : recovering quasi-structured meshes. 51st AIAA Aerospace Sciences Meeting, Jan 2013.

[120] W.C.L. Shih, C. Wang, D. Coles, and A. Roshko. Experiments on flow past rough circular cylinders at large reynolds numbers. J. Wind Eng. Indust. Aerodyn., 49 :351–368, 1993.

[121] C.W. Shu and B. Cockburn. Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems. J. Sci. Comput., 16(3) :173–261, 2001.

[122] C.W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys., 77 :439–471, 1988.

[123] M.L. Shur, M.K. Strelets, A.K. Travin, and P.R. Spalart. Turbulence modeling in rotating and curved channels : Assessing the Spalart-Shur correction. AIAA Journal, 38(5) :784–792, 2000.

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

[125] P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows. In 30th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-92-0439, Reno, NV, USA, Jan 1992.

[126] P.R. Spalart, S. Deck, M. Shur, K. Squires, M. Strelets, and A. Travin. A new version of detached-eddy simulation, resistant to ambiguous grid densities. Theoretical and Computational Fluid Dynamics, 20 : :181–195, 2006.

[127] P.R. Spalart, S. Deck, M. Strelets, 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., 20 :181–195, 2006.

[128] P.R. Spalart and M.L. Shur. On the sensitization of turbulence models to rotation and curvature. Aerospace Science and Technology, 5 :297–302, 1997.

[129] A. S. Stabnikov and A. V. Garbaruk. J. Phys. : Conf. Ser., 1697 :012224, 2020.

[130] S. Toosi and J. Larsson. Towards systematic grid selection in LES : identifying the optimal spatial resolution by minimizing the solution sensitivity. Computers and Fluids, doi :https ://doi.org/10.1016/j.compfluid.2020., 201 :104488, 2020.

[131] S. Toosi and J. Larsson. The Germano identity error and the residual of the LES governing equation. Journal of Computational Physics, 443 :110544, 2021.

[132] A. Vreman, B. Geurts, and H. Kuerten. A priori tests of large-eddy simulation of compressible plane mixing layer. Journal of Engineering Mathematics, 29 :299–327, 1995.

[133] P. Wen and W. Qiu. Investigation of drag crisis phenomenon using cfd methods. Applied Ocean Research, 67 :306–321, 2017.

[134] J. Wu, J.Z. Zhu, J. Szmelter, and O.C. Zienkiewicz. Error estimation and adaptivity in Navier-Stokes incompressible flows. Computational Mechanics, 6 :259–270, 1990.

[135] G. Yan and C.F. Ollivier-Gooch. Accuracy of discretization error estimation by the error transport equation on unstructured meshes - nonlinear systems of equations. In 22, AIAA Paper2015-2747, Dallas, TX, USA, Jun 2015.

[136] G. Yan and C.F. Ollivier-Gooch. Accuracy of Discretization Error Estimation by the Error Transport Equation on Unstructured Meshes - Nonlinear Systems of Equations. American Institute of Aeronautics and Astronautics, 2017/11/09 2015.

[137] G. Yan and C.F. Ollivier-Gooch. Towards higher order discretization error estimation by error transport using unstructured finite-volume methods for unsteady problems. Computers and Fluids, 154 :245–255, 2017.

[138] S.M. Yeon, J. Yang, and F. Stern. Large-eddy simulation of the flow past a circular cylinder at sub– to super–critical Reynolds numbers. Applied Ocean Research, 59 :663–675, 2016.

[139] S. Yokuda and R.R. Ramaprian. The dynamics of flow around a cylinder at subcritical Reynolds number. Phys. Fluids A, 5 :3186–3196, 1990.

[140] M.M. Zdravkovich. Flow around circular cylinders. Vol 1 : Fundamentals. Oxford Science Publications. Oxford University Press, 1997.

[141] Y. Zhang, M.M.Rahman, and G. Chen. Development of k-R turbulence model for wall-bounded flows. Aerospace Science and Technology, 98, 2020.