Mesh adaptation for elliptic equations on quadtree/octree grids

Lucas Prouvost

To cite this version:

Lucas Prouvost. Mesh adaptation for elliptic equations on quadtree/octree grids. Fluid mechanics [physics.class-ph]. Sorbonne Université, 2022. English. NNT : 2022SORUS392 . tel-03966961

HAL Id: tel-03966961

https://theses.hal.science/tel-03966961v1

Submitted on 1 Feb 2023

Sorbonne Université

Ecole Doctorale 391 : Sciences Mécaniques, Acoustique, Electronique et Robotique

Thèse de Doctorat

Spécialité Mécanique

Présentée par
Lucas Prouvost

Pour le titre de
Docteur de Sorbonne Université

Mesh adaptation for elliptic equations on quadtree/octree grids

Sous la direction de
Daniel Fuster (CNRS)

et le co-encadrement de
Anca Belme (MCF)

à l'Institut Jean le Rond ∂'Alembert, Sorbonne Université, CNRS, UMR 7190

Devant le jury composé de :

M. Adrien Loseille	Chargé de recherche INRIA	Rapporteur
M. Franck Ledoux	Directeur de recherche CEA	Rapporteur
M. Vincent Le Chenadec	Maître de conférence Université Gustave Eiffel	Examinateur
M. Stéphane Popinet	Directeur de recherche CNRS	Président du jury
M. Jean-Camille Chassaing	Professeur Sorbonne Université	Invité
Mme. Anca Belme	Maître de conférence Sorbonne Université	Co-encadrante
M. Daniel Fuster	Directeur de recherche CNRS	Directeur

Abstract

Finding an efficient and fully automated mesh adaptation method is nowadays one of the most challenging and important research subject in Computational Fluid Dynamic (CFD), as it allows to optimize the solution accuracy in regard to the running time of simulations. In this thesis, we develop a new Adaptive Mesh Refinement (AMR) method for the CFD software Basilisk.

After introducing the continuous mesh framework, a continuous representation of a discrete mesh in Riemannian metric spaces, we extend the metric-based interpolation error to the quad/octree grid structure of Basilisk and its square/cubic elements. By solving an optimization problem, an estimation of the minimal interpolation error is obtained. It is later used as a reference to assess the AMR method performances.

When solving elliptic equations, such as the Poisson-Helmholtz equation, we show that the minimization of the interpolation error does not guarantee the minimization of the numerical error. The role of the mesh compression – the size aspect ratio between the finest cell size and the mean cell size of an adapted mesh – is observed and shows that a slightly sub-optimal mesh toward the interpolation error may significantly reduces the total error of a numerical solution. From that, the classical user-imposed minimum element size criteria set to avoid over-refinement is replaced by an automatic estimation of the minimal element size, which acts as an additional constrain in the adaptation method. It is used in combination with an error estimation which assumes that the interpolation error propagates in a small neighbourhood around each mesh elements due to the discretization of the equation. The efficiency of the method is assessed on several test-cases.

Finally, based on the observation that the incompressible Navier-Stokes solver of Basilisk relies on the resolution of an elliptic equation, an extension of the error estimate is proposed for the incompressible Navier-Stokes equations. The obtained AMR method is tested on several flow configurations, for flow fields submitted to various Reynolds numbers.

Résumé

De par sa capacité à minimiser les erreurs et le temps de calcul des simulations numériques, l'adaptation de maillage est l'un des enjeux majeurs de la recherche actuelle. Dans cette thèse, nous proposons une nouvelle méthode d'adaptation de maillage appliquée au logiciel Basilisk.

Après avoir introduit la notion d'espace métrique riemannien, qui permet de définir un équivalent continu aux maillages discrets, nous adaptons l'estimation de l'erreur d'interpolation basée métrique aux maillages *quad/octree* de Basilisk. En resolvant un problème d'optimisation sous contrainte, nous estimons l'erreur d'interpolation minimale qu'il est possible d'obtenir sur ces maillages, et nous l'utilisons par la suite comme critère de référence pour évaluer la performance des méthodes d'adaptation de maillage.

Dans le cas de la résolution numérique d'équations elliptiques, telle l'équation de Poisson-Helmhotlz, nous montrons que minimiser l'erreur d'interpolation n'est pas suffisant pour minimiser l'erreur numérique. Le taux de compression du maillage adapté – le rapport entre la taille de ses plus petits éléments et la taille moyenne de ses éléments – joue un rôle important. En particulier, il permet de montrer qu'un maillage légèrement sous-optimal en terme d'erreur d'interpolation peut réduire de façon significative l'erreur totale mesurée sur une solution numérique. En se basant sur ce constat, nous proposons une estimation automatique de la taille minimale d'un maillage adaptée. Elle s'ajoute à la méthode d'adaptation comme une contrainte supplémentaire. En association avec ceci, nous présentons un estimateur d'erreur pour l'équation de Poisson-Helmholtz qui considère que l'erreur totale d'une solution numérique provient d'une propagation locale de l'erreur d'interpolation. La méthode d'adaptation ainsi obtenue est validée sur quelques cas-tests.

Enfin, le solveur de Basilisk resolvant les équations de Navier-Stokes dans le cas incompressible repose sur la résolution d'une équation elliptique. Nous proposons donc une extension à ce cas de notre estimateur d'erreur. La méthode d'adaptation dans son ensemble est testée sur différents écoulements à différents nombres de Reynolds.

Contents

1 Introduction **1**
- 1 Context . . . 1
  - 1.1 Physical and numerical challenges . . . 1
  - 1.2 Mesh adaptation . . . 4
  - 1.3 Mesh quality . . . 9
- 2 Content of this thesis . . . 10

2 Basilisk: a software for partial differential equation solutions **13**
- 1 Quad/octree grid structure . . . 14
- 2 Numerical solvers . . . 16
  - 2.1 Poisson-Helmholtz solver . . . 17
  - 2.2 Incompressible Euler/Navier-Stokes solver . . . 18
- 3 Wavelet-based AMR method . . . 20
- 4 Limits of the wavelet-based AMR method . . . 22
- 5 Conclusion . . . 25

3 Interpolation error model **27**
- 1 Riemannian metric space for mesh adaptation . . . 28
  - 1.1 Euclidean and Riemannian metric space . . . 28
  - 1.2 Unit mesh . . . 30
- 2 The continuous mesh framework . . . 31
  - 2.1 Duality discrete-continuous mesh . . . 31
  - 2.2 Continuous linear interpolate and continuous interpolation error . . . 33
- 3 Continuous interpolation error for quad/octree grids . . . 35
  - 3.1 Quadtree continuous interpolation error . . . 35
  - 3.2 Octree continuous interpolation error . . . 38
  - 3.3 Uniformly refined mesh interpolation error . . . 39
- 4 Optimal metric-based mesh adaptation . . . 41
  - 4.1 $\mathbf{L}^{\mathbf{p}}$-norm anisotropic metric-based interpolation error . . . 41
  - 4.2 $\mathbf{L}^{\mathbf{p}}$-norm isotropic interpolation error for quad/octree grids . . . 43
  - 4.3 Adaptive Algorithm . . . 45
- 5 Quad/Octree optimal error validation . . . 46
  - 5.1 Multiscale example . . . 46
  - 5.2 Boundary layer example . . . 49
- 6 Conclusion . . . 50

4 A novel AMR method for elliptic equations 53
- 1 Numerical error quantification for elliptic equations 54
  - 1.1 Experiment and first observations: a coarsening study 54
  - 1.2 Definition of AMR performance quantities 56
- 2 A simple error model for PDE equations 63
- 3 Mesh compression ratio constrain 64
  - 3.1 Estimation methods for the compression ratio constrain 65
  - 3.2 Security coefficient 72
- 4 Proposed AMR strategy 74
  - 4.1 AMR algorithm 74
  - 4.2 Illustrative example 74
- 5 Validation 76
  - 5.1 Boundary layer 77
  - 5.2 Multiscale sinus 79
  - 5.3 Potential flow around a circular cylinder 82
- 6 Conclusion 84

5 An AMR method for the solution of the incompressible Navier-Stokes equations 85
- 1 A simple error model for the Navier-Stokes equations 86
- 2 Application of AMR to different flows 86
  - 2.1 Lamb-Oseen vortex 87
  - 2.2 Flow around a circular cylinder 89
  - 2.3 Lid-driven cavity 104
- 3 Conclusion 106

6 Conclusion and perspectives 107

A Error on a square element of reference 111

B Optimal quad/octree interpolation error proof 113

C Problems of reference 117
- 1 Multiscale sinusoidal problem 117
- 2 Boundary layer problem 118

D Poisson-Helmholtz solver – A refining study 121

E Lamb-Oseen vortex additional results 123

Bibliography 125

List of Figures

1.1 Example of a mesh (black triangles) discretizing a complex surface. 2

1.2 Mesh adaptation principle applied to the field of image compression. On top, from left to right, three finer and finer meshes are represented. Below, an image obtained on the corresponding mesh is presented, where, as a pixel, each mesh element contains one color. The left picture has visibly a low accuracy in comparison with the two others. On the other side, the right mesh is so fine that the cells can't be distinguish. The middle mesh seems to provide a reasonably accurate picture for a reasonable number of element. Thus, it is a well adapted mesh. In the context of CFD, the image represents the solution of a PDE. 2

1.3 Experimental setup of the CACHMAP project [8]. 3

1.4 Representation of the interpolation error and the total error (light gray). . . . 4

1.5 Representation of the order of approximation of 1D elements. In this example, the interpolation (blue) of the solution (red) is exact at vertex (black points) and additional nodes (black circles). The interpolation error is represented by the gap between the solution and its interpolation (light gray). This error is reduced when the order of the interpolation increases. 6

1.6 Examples of adapted meshes. 7

1.7 Mesh before (bad quality elements) and after (better quality elements) gradation associated with their prescribed metric fields. (images extracted from [80]) 10

1.8 30 000 elements meshes comparison. Left: the ratio between the minimal and the mean cell size is not constrained. Right: this ratio is constrained. As a consequence, the element size variation is reduced. 10

2.1 A quadtree grid structure corresponding to meshes on figures 2.2 and 2.3. The mesh elements (*ie.* the leaf cells) are represented with gray triangle. Parent cells are represented with black dots. The level 1 cells are the *children* of the level 0 cell. The level 0 cell is the *parent* of the level 1 cells. 14

2.2 A quadtree mesh corresponding to the quadtree structure represented on figures 2.1 and 2.3. 15

2.3 A 3D representation of the tree-structure of the mesh illustrated on figures 2.1 and 2.2. Leaf cells are colored by level in gray (from lighter to darker gray, level 1 to 3). Parent cells are not colored. The level 0 cell is the root cell. . . 15

2.4 Examples of meshes respecting (or not) the Basilisk requirements. The neighboring cells which do not respect Basilisk mesh constraints are filled in red. . 16

2.5 Local 3×3 stencil around a cell indexed $[0, 0]$. Face and neighbours matrix-like indices notations are shown. Faces (gray indices) are indexed with $\pm\frac{1}{2}$ and neighbouring cells (black indices) with ± 1. In practice, a 5×5 Cartesian stencil is defined in Basilisk. 17

2.6 Principle of the *wavelet-based* AMR method (figure adapted from [26]). (a) The field $u(\boldsymbol{x}_c^l)$ (black dots) is downsampled to $u(\boldsymbol{x}_c^{l-1})$ (blue dots), which is then upsampled (b) by linear interpolation to $\left(\Pi u^{l-1}\right)(\boldsymbol{x}_c^l)$ (red dots). (c) The error is estimated as $|u(\boldsymbol{x}_c^l) - \left(\Pi u^{l-1}\right)(\boldsymbol{x}_c^l)|$. 21

2.7 Representation of the function $u_1(x, y)$. 23

2.8 Mesh and level field obtained for three adaptive criterion thresholds. From left to right, the mesh is coarsen. 24

2.9 $\boldsymbol{L}^1$-norm error *vs.* square root of number of cells. The error is measured on the numerical solution of the partial differential equation (2.23). Red dots and line represent the error convergence on a sequence of uniformly refined meshes. Green triangles represent the error convergence on adapted grids with minimum level $l = 11$. The numbers correspond to the meshes represented on figure 2.8 25

3.1 Left: a 2D unit ball associated with a metric $\mathcal{M}$. Right: natural mapping between the unit balls of a metric space $(\mathbb{R}^n, \mathcal{M})$ and the physical space $(\mathbb{R}^n, I_n)$. $h_i = \lambda_i^{-\frac{1}{2}}$ are the eigenvalues of $\mathcal{M}^{-1/2}$. 29

3.2 Examples of unit tetrahedra and triangles for a given metric represented by its unit ball (figure extracted from [122]). 32

3.3 Conventions used to enumerate the edges and the faces of a triangle and of a tetrahedron. (figure extracted from [122]) 32

3.4 Partition of the square element K into two triangles K_1 and K_2. 36

3.5 Comparison between the error measured on uniform meshes for a function and its theoretical prediction. 40

3.6 L^p-norm global error of the multiscale sinusoidal function u_1 *vs* square root of cell number (inversely proportional to the mean cell size). These error convergence curves compare the results obtained with $\boldsymbol{L}^p$ isotropic AMR method and theory, wavelet-based Basilisk AMR method, and uniformly refined meshes. Left: $\boldsymbol{L}^1$ norm AMR. Right: $\boldsymbol{L}^2$ norm AMR. 47

3.7 Mesh and level field obtained for three AMR for the multiscale sinusoidal function u_1: $\boldsymbol{L}^1$-norm isotropic metric-based meshes, $\boldsymbol{L}^2$-norm isotropic metric-based meshes, and wavelet-based Basilisk adapted meshes. Each mesh contains around 600 000 cells. 48

3.8 L^p-norm global error of the boundary layer function u_2 *vs* square root of cell number (inversely proportional to the mean cell size). These error convergence curves compare the results obtained with $\boldsymbol{L}^p$ isotropic AMR method and theory, wavelet-based Basilisk AMR method, and uniformly refined meshes. Left: $\boldsymbol{L}^1$ norm AMR. Right: $\boldsymbol{L}^2$ norm AMR. 49

3.9 Mesh and level field obtained for three AMR for the boundary layer function u_2: $\boldsymbol{L}^1$-norm isotropic metric-based meshes, $\boldsymbol{L}^2$-norm isotropic metric-based meshes, and wavelet-based Basilisk adapted meshes. Each mesh contains around 600 000 cells. 50

4.1 Illustration of the coarsening study: from the initial uniform mesh (a), the cells with the least interpolation error are coarsen. This operation is repeated and provides successively the mesh (b) and (c). 54

4.2 Total error (blue) obtained for a coarsening study: an initially uniform mesh (red) is successively coarsen based on the interpolation error. This is repeated from several uniform meshes (Left). Right: two main error regimes A and C are observed, separated by a regime of transition B, which shows performances similar to the theoretical optimum (black). The error in the regime C becomes larger than the error on uniform meshes. 55

4.3 Compression ratio η_N, loss factor L and gain factor G visualisation for two different N elements meshes A and B having they own minimal cell size $h_{min}(A)$ and $h_{min}(B)$. 57

4.4 Loss factor *vs* compression ratio representation for the coarsening study. Blue: loss factor representative for the total error of the numerical solution. Red: loss factor representative for the interpolation error. The curves for the different coarsening studies ($l_{max} \in [\![8, 11]\!]$) superpose each other. For the total error, an almost optimal loss factor $L \approx 1.4$ is obtained for the corresponding optimal compression ratio $\eta_{N,opt} \approx 0.11$. 58

4.5 Meshes obtained for two compression ratio and for different minimum cell size. 59

4.6 Normalized distributions for the meshes represented in figure 4.5. The distributions for the two $\frac{\eta_N}{\eta_{N,opt}} = 1$ meshes (light and dark blue) are identical. They are different from the distributions of the two $\frac{\eta_N}{\eta_{N,opt}} = 0.3$ meshes (light and dark red). 60

4.7 Top and middle: Local numerical error *vs* local interpolation error for varying compression ratio η_N and maximum level of refinement l_{max}. Bottom: Coefficient of determination r^2 of the linear fit for all of the meshes of the coarsening study. For $\frac{\eta_N}{\eta_{N,opt}} \geq 1$ both errors are linearly correlated. For $\frac{\eta_N}{\eta_{N,opt}} < 1$, they are completely decorrelated. 62

4.8 Mesh compression comparison. 64

4.9 Representation of the approximated optimal compression ratio $\eta_{N,opt}$ and the associated gain factor G_{opt} on a mesh with the minimal cell size $h = \frac{1}{512}$. The numerical results come from the coarsening study in section 1. 66

4.10 Schematic representation of the estimation of the optimal compression ratio $\eta_{N,opt}$ and the corresponding gain factor G. 67

4.11 Steps to estimate $\eta_{N,\,opt}$ using the virtual AMR algorithm. 68

4.12 Example of virtual meshes respecting a given error criterion. 69

4.13 Comparison between the optimal compression ratio provided by the metric-based theory η_{metric} and the tree-based algorithm η_{tree}. They are compared with the loss factor *vs* compression ratio curves obtained from coarsening studies based on the numerical solution (see section 1). 71

4.14 Schematic representation of the influence of a compression ratio constrain η_ξ. The different colored regions correspond to different efficiency of the meshes. (C): non-efficient adapted mesh in terms of interpolation error. (D): no mesh. (A) and (B): adapted meshes which reduce the interpolation error in comparison with uniformly refined meshes. Meshes in (A) respect $\eta \geq \eta_N$. The total error in (B) can be larger than the total error on a cartesian mesh. 73

4.15 Result of the mesh optimization algorithm that minimises the interpolation error (bottom left) and the grid obtained when imposing the constrain on the minimum grid size (bottom right). The corresponding total error convergence is presented (top). 76

4.16 AMR method performance evaluation for the solution (equation (C.3)) of the Poisson-Helmholtz equation (4.34). The current AMR method performs closely to the theoretical optimum with a security coefficient $\xi = 0.05$, and less closely to the optimal with $\xi = 0.3$. The AMR without compression ratio ($l_{max} = 15$) is near optimal. 77

4.17 Comparison between $\boldsymbol{L}^1$ and $\boldsymbol{L}^2$ AMR for the numerical boundary layer solution (equation (C.3)). 79

4.18 AMR method performance evaluation for the solution (equation (C.1)) of the Poisson-Helmholtz equation (4.34) for two AMR methods: one imposing $l_{max} = 15$ (empty blue dots), the other based on the optimal compression ratio and imposing $l_{max}(\eta_{\xi=0.3})$ (filled blue dots). 80

4.19 Performances comparison between the current AMR method (blue dots) and the wavelet-based method of Basilisk (green dots and circles) for the multiscale sinusoidal numerical solution. The wavelet-based method is sensitive to the authorized maximum level. The current AMR method performs better (less error and loss factor for a given number of elements). 81

4.20 Comparison between $\boldsymbol{L}^1$ and $\boldsymbol{L}^2$ AMR for the numerical multiscale sinusoidal solution (equation (C.1)). 82

4.21 Error convergence of the current AMR method for the cylinder in a potential flow. 83

4.22 Meshes obtained for the cylinder in a potential flow. 83

5.1 Zoom in $[-1, 1]^2$ of the solution: pressure field and velocity vector field. . . . 87

5.2 Error convergence on the pressure obtained for the Lamb-Oseen vortex solution of the Euler equations. 88

5.3 Adapted mesh levels obtained for the Lamb-Oseen vortex-like solution of the Euler equations in the domain $[-5, 5] \times [-5, 5]$. 89

5.4 Pressure field and streamlines for a flow around a circular cylinder at Reynolds $Re = 0.05$. Zoom in the range $[-10, 10] \times [-10, 10]$. 91

5.5 Enstrophy convergence for a flow around a circular cylinder at Reynolds $Re = 0.05$. Left: Enstrophy convergence. Right: Error convergence. Top: influence of the security coefficient. Bottom: comparison with the wavelet-based AMR method. 92

5.6 Meshes obtained for the flow around a circular cylinder at Reynolds $Re = 0.05$ with the proposed AMR method with $\xi = 0.1$ (left, $N = 15\,016$ elements), with $\xi = 0.8$ (middle, $N = 15\,004$ elements) and with the wavelet-based AMR method (right, $N = 15\,037$ elements). Top: zoom in the range $[-10, 10] \times [-10, 10]$. Bottom: complete domain $[-64, 64] \times [-64, 64]$. 93

5.7 Drag coefficient convergence for a flow around a circular cylinder at Reynolds $Re = 0.05$. Left: influence of the security coefficient. Right: comparison with the wavelet-based AMR method. 93

5.8 Pressure field and streamlines for a flow around a circular cylinder at Reynolds $Re = 0.5$. Zoom in the range $[-10, 10] \times [-10, 10]$. 94

5.9 Enstrophy convergence for a flow around a circular cylinder at Reynolds $Re = 0.5$. Left: Enstrophy convergence. Right: Error convergence. Bottom: influence of the security coefficient. Top: comparison with the wavelet-based AMR method. 95
5.10 Drag coefficient convergence for a flow around a circular cylinder at Reynolds $Re = 0.5$. Left: Drag coefficient convergence. Right: Error convergence. Top: influence of the security coefficient. Bottom: comparison with the wavelet-based AMR method. 96
5.11 Meshes obtained for the flow around a circular cylinder at Reynolds $Re = 0.5$ with the proposed AMR method with $\xi = 0.1$ (left, $N = 14\,986$ elements), with $\xi = 0.8$ (middle, $N = 14\,992$ elements) and with the wavelet-based AMR method (right, $N = 11\,581$ elements). Top: zoom in the range $[-10, 10] \times [-10, 10]$. Bottom: complete domain $[-64, 64] \times [-64, 64]$. 97
5.12 Pressure field and streamlines for a flow around a circular cylinder at Reynolds $Re = 40$. Zoom in the range $[-10, 10] \times [-10, 10]$. 98
5.13 Enstrophy convergence for a flow around a circular cylinder at Reynolds $Re = 40$. Left: Enstrophy convergence. Right: Error convergence. Top: influence of the security coefficient. Bottom: comparison with the wavelet-based AMR method. 99
5.14 Drag coefficient convergence for a flow around a circular cylinder at Reynolds $Re = 40$. Left: influence of the security coefficient. Right: comparison with the wavelet-based AMR method. 100
5.15 Meshes obtained for the flow around a circular cylinder at Reynolds $Re = 40$ with the proposed AMR method with $\xi = 0.1$ (left, $N = 14\,992$ elements), with $\xi = 0.8$ (middle, $N = 14\,995$ elements) and with the wavelet-based AMR method (right, $N = 12\,250$ elements). Top: zoom in the range $[-10, 10] \times [-10, 10]$. Bottom: complete domain $[-64, 64] \times [-64, 64]$. 100
5.16 Pressure field and streamlines for a flow around a circular cylinder at Reynolds $Re = 100$ at time $t_{end} = 300$. Zoom in the range $[-10, 20] \times [-10, 10]$. 101
5.17 Top: enstrophy evolution in time. Bottom: mean enstrophy convergence. . . 102
5.18 Top: drag coefficient evolution in time. Bottom: mean drag coefficient convergence. 103
5.19 200 000 elements mesh for the flow around a circular cylinder at Reynolds $Re = 100$. 103
5.20 Pressure field and streamlines for the lid-driven cavity at Reynolds $Re = 1\,000$. 104
5.21 Comparison between the velocity profiles $\boldsymbol{u} = (u, v)^T$ obtained for three $N \approx 2\,800$ elements adapted meshes and the reference. The compression ratio-based constrained methods ($l_{max}(\eta_{\xi=0.1})$) matches the reference profiles. The same AMR procedure without constrain ($l_{max} = 12$) and the wavelet-based methods are less accurate. 105
5.22 Comparison between the meshes obtained with the compression ratio-based and the wavelet-based methods for a lid-driven cavity at Reynolds $Re = 1\,000$. 106

C.1 Representation of the function $u_1(x, y)$. 117
C.2 Representation of the function $u_2(x, y)$. 118

D.1 Error convergence for the refining study: a coarse mesh (level $l = 5$) is refined where the interpolation error on the multiscale sinusoidal numerical solution is the highest. 121

E.1 error on $\boldsymbol{u}$ for the Lamb-Oseen vortex. 123

LIST OF TABLES

3.1 Comparison between the theoretical error prefactor and the measured error prefactor for different functions. 41
3.2 Error in L^1, L^2 and L^∞ norm on the three meshes figure 3.7 for u_1. 48

4.1 Interpolation and total error comparison for the meshes represented in figure 4.1. Two different meshes may have a comparable interpolation error, but a huge gap between their total error. 60

5.1 Regimes of flows around a circular cylinder [134, 135] categorized by their Reynolds number Re. 89
5.2 Number of elements obtained at the end of each AMR simulation for a circular cylinder at Reynolds $Re = 100$. 101

1| Introduction

1 Context

1.1 Physical and numerical challenges

The numerical resolution of physical and mechanical problems is of primordial importance in numerous applications, from academical purposes such as water-surface bursting bubbles [1] to industrial conceptions such as planes [2]. It is common knowledge that the analytical solution of the mathematical equations governing these complex and potentially multi-scale phenomena is out of reach, while building and testing experimental prototypes is a highly costly operation. Numerical simulations render feasible the understanding and improvement of complex physical phenomena making them a key ingredient in the decision making processes. However, the numerical resolution of equations in problems where multiple scales are involved require a huge amount of time to provide accurate solutions. Thus, a single computation can be too costly and time consuming, especially for industries and their competitive environment. For example, a simulation of the onset of the second stage of labor can take more than 3 months on thousands of cores [3].

The hardware performance is of primary importance in order to reduce the simulation time. But the miniaturisation of transistors and microprocessors, which helps improving their efficiency, tends to reach a size limit due to quantum effects. Thus, improving the hardware performance alone is not sufficient, and the software performance – mainly the numerical solvers and meshing software – is of huge importance.

A common methodology to increase the efficiency of computations is the use of adaptive methods, as for example automatic adaptive meshes. A mesh is a discrete representation of a computational domain in a finite number of (usually tetrahedral or hexahedral) elements, such as illustrated in figure 1.1. A clever solution is to increase locally the mesh resolution in subdomains where the solution is less accurate (or precise). In the context of image compression, an illustration of the benefit of using mesh adaptation techniques by adding mesh elements where the image color contrast is more important can be seen in figure 1.2. This illustration clearly shows that it is much more efficient to add mesh elements where they are needed than refining the whole picture. In Computational Fluid Dynamic (CFD) applications, the more elements a mesh has the more precise the simulation results should be, but the computational cost increases drastically since the numerical solution is often computed on each mesh element (a cell). Mesh adaptation is then an interesting alternative for multi-scale, multi-phase flows or when the flow field presents singularities. The search of efficient and fully automated adaptation methods is under continuous development and has been identified as one of the major bottleneck in CFD workflows in the CFD vision 2030 Study [4, 5].

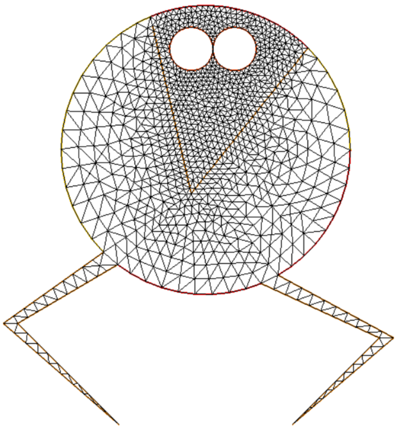
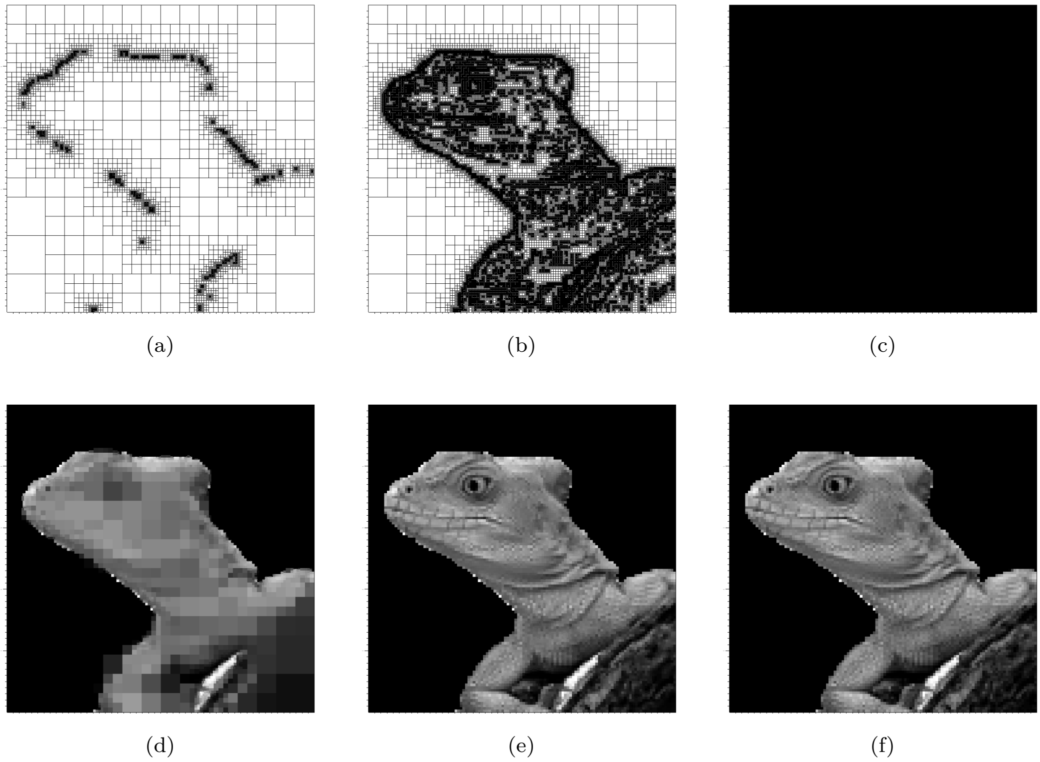


Figure 1.1: Example of a mesh (black triangles) discretizing a complex surface.



(a) (b) (c)

(d) (e) (f)

Figure 1.2: Mesh adaptation principle applied to the field of image compression. On top, from left to right, three finer and finer meshes are represented. Below, an image obtained on the corresponding mesh is presented, where, as a pixel, each mesh element contains one color. The left picture has visibly a low accuracy in comparison with the two others. On the other side, the right mesh is so fine that the cells can't be distinguish. The middle mesh seems to provide a reasonably accurate picture for a reasonable number of element. Thus, it is a well adapted mesh. In the context of CFD, the image represents the solution of a PDE.

The objective of this thesis is to improve the Adaptive Mesh Refinement (AMR) method of the software Basilisk [6]. This is a first step toward an efficient AMR method for multi-phase compressible flows, such as encountered in the ANR project CACHMAP [7, 8] and its successor (ANR ProBalCav). These projects aim at describing, optimizing and conceiving new cavitating multilayered materials: these assemblies are simple sandwiches composed with one liquid layer surrounded by two solid layers, as represented figure 1.3. When a shock wave propagates through this material, complex phenomena come into play (formation of bubbles, shock-bubble interactions, ...). For such a complex physical problem, it is mandatory to use numerical simulations to help understand it and to access results which are not obtainable through experimental studies. However, the full numerical simulation of these phenomena is a real challenge as it mixes multiphase flows, compressible effects and even possible interactions between the fluids and the surrounding structure. This implies a complex fluid flow problem (eventually coupled with the structure) that requires advanced numerical tools to be solved numerically. The use of an efficient automated AMR method is then essential.

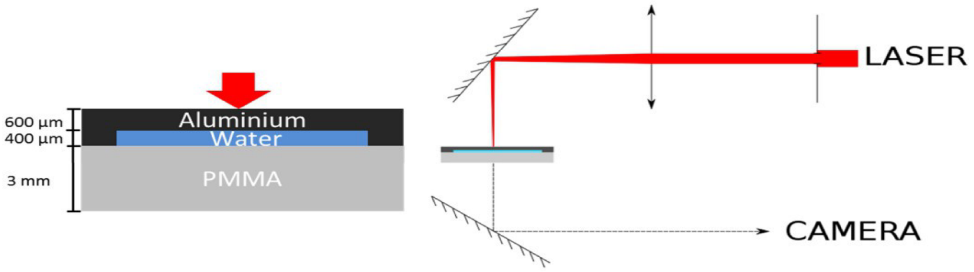


Figure 1.3: Experimental setup of the CACHMAP project [8].

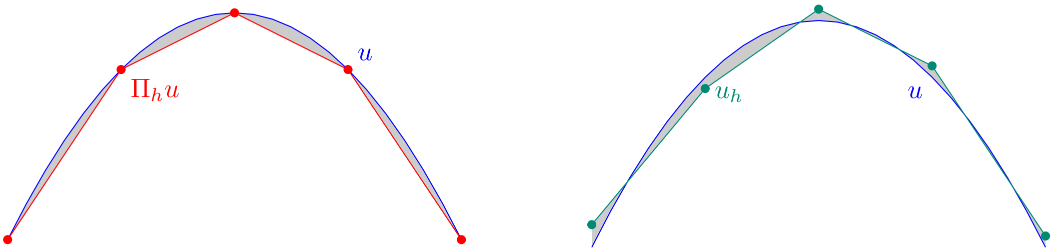
The main purpose of mesh adaptation is to reach an optimal balance between the computational cost and the numerical error reduction (accuracy). The accuracy of a numerical solution is affected by errors coming from several sources. In particular, in CFD, we distinguish at least [9, 10, 11, 12]:

- Modeling errors. We often use mathematical models to describe realistic physical phenomena without necessary assessing the errors committed by the underlying assumptions of the models.

- Geometry errors. They directly derive from the discrepancies between the numerical and true design.

- Discretization errors (or total error). They account for the errors due to the discrete representation (mesh, equation, ...) of a continuous solution and problem (see figure 1.4). In particular, we talk about interpolation errors for whose coming from the discretization of the domain into elements of size h. The interpolation errors affect only locally the solution accuracy. On the opposite, the numerical errors, which depend on the numerical schemes and their implementation (equation discretization), are not necessarily local: they can be advected or diffused from an element of the mesh to the entire numerical domain.

The first two errors concern mainly the validity of the numerical model and its geometry and are out of the scope of this thesis. We concentrate on minimising the errors introduced when projecting the solution on a discrete mesh and when solving a discrete set of equations. It is a usual practice to quantify mesh discretization errors using interpolation errors since they are local by nature and can be used as indicator for mesh refinement. However, it is *a priori* not assured that minimizing interpolation errors really minimizes the total numerical errors.



(a) Interpolation error between the continuous solution u and its linear interpolation $\Pi_h u$.

(b) Total error between the continuous solution u and the numerical solution u_h.

Figure 1.4: Representation of the interpolation error and the total error (light gray).

1.2 Mesh adaptation

1.2.1 General description

As previously said, a mesh is the spatial discretization of the physical domain of interest. This discretization forms a partition of the domain. Locally, a mesh is composed by elements (or cells) characterized by their faces, edges and vertices. The link between the elements (set of neighbours, shared vertex, ...) is called the connectivity of the elements. Meshes can be classified through various structures and element shapes.

A *structured mesh* presents a regular distribution of elements. Each element is characterized by a set of index (i, j, k), which implicitly provides the full knowledge of the element connectivity[1]. The most classical structured mesh contains square elements in 2D and cubic elements in 3D.

On the opposite, an *unstructured mesh* does not have an implicit connectivity pattern. The elements are defined by their nodes and surfaces, and they need to be linked by a connectivity table. Many unstructured meshes are composed by simplex (triangular or tetrahedral elements), but any form of element can be used in such a mesh, and different element may even be mixed. The most common mixed elements meshes contains tetrahedra and hexahedra. The choice of the type of element which compose a mesh is critical regarding the performance of the meshing software and the numerical solution accuracy: the generation of meshes composed with high quality tetrahedra is more robust and automatic than the generation of full-hexahedral meshes, but the use of meshes containing hexahedra generally increase the speed and the accuracy of the numerical resolutions [13, 14].

An *adapted mesh* allows local variations in size and/or orientation of the elements based on a given criterion. The idea behind these variations is to automatically concentrate the computational costs in regions which are critical to capture accurately the flow features, and to alleviate them elsewhere. Globally, this should either improve the accuracy of the solution without excessive increase of the computational costs or reduce the computational efforts

---

[1] The elements $(i-1, j, k)$ and $(i+1, j, k)$ are respectively the left and right neighbours of the element (i, j, k). The neighbouring elements in the other directions are similarly described using the indices j and k.

without loss in the solution accuracy.

The generation of a mesh is often performed by a meshing software such as *feflo* [15], *mmg3d* [16] or *gmsh* [17]. Each of them has its own specificities (type of elements, underlying meshing method, ...). In the context of full-hexahedral meshes, the search for a robust and automatic method to mesh arbitrary volumes with high quality elements remains an unsolved challenge, which has seen the development of numerous methodologies (advancing front methods, grid-based methods, ...) [18]. Among them, the *frame-fields* methods [13, 14], which solves a constrained optimization problem to find the local orientation of the hexahedra seems particularly promising. In this thesis, we focus on *tree* grid structures [19, 20, 21]. This type of method is one of the most robust [18] and is the type of grids used in the software Basilisk. They allow to perform efficient mesh adaptation methods by dividing previously existing elements into sub-elements. The succession of nested elements obtained from the elements divisions forms an underlying tree structure which governs the elements connectivity. If any type of element could theoretically constitute this type of mesh, many tree grid structures are composed with quadrilateral elements or hexahedra. Basilisk quad/octree meshes contain only square/cubic elements and the mesh structure is fully integrated in the core of the software. Its specificities are described in chapter 2.

1.2.2 Adaptive Mesh Refinement (AMR) methods

Adaptive methods are usually distinguished into three categories – p-adaptation, r-adaptation and h-adaptation methods – which are briefly described hereafter.

P-adaptation [22] is mainly associated with Finite Element discretizations. In this type of adaptation, starting from an initial mesh, the number of mesh elements and the position of the vertex remain constant. However, a solution is represented by polynomials on the elements of the mesh, as illustrated in figure 1.5. The local accuracy of the solution depends on the order of these polynomial functions. Thus, the use of higher-order polynomials in well-chosen portion of a domain leads to a general improvement of the solution accuracy. P-adaptation consists to locally adapt the order of the elements of the mesh. For problems with sufficient smoothness, it can achieve higher convergence rates than h-adaptation methods [22]. However, a special care is needed to create numerical schemes which tolerate this mix of elements with different orders. An example of p-adapted mesh is shown in figure 1.6c.

The r-refinement methods consist in the modification of the spatial resolution of an existing mesh [23, 24]. This change of resolution is made at constant number of nodes and cells and at unchanged mesh connectivity. That means that all the mesh elements and vertex keep exactly the same neighbours: they are simply displaced from their original position, as illustrated in figure 1.6d. The main advantage of this type of method is that the computational cost remains almost constant throughout the simulations, as the number of elements is constant. Nonetheless, the overall accuracy of the solution is limited by the initial number of elements.

H-adaptation methods are the most widespread adaptive methods for Finite Volume discretizations. They are also called Adaptive Mesh Refinement (AMR) methods. Their core concept is to adapt the size of the mesh elements: fine elements are required in critical regions, whereas coarse elements are sufficient elsewhere. They allow efficient mesh structures or mesh generations, but they can lead to a high increase in the number of degree of freedom of a problem. The creation of new elements in h-adaptation techniques may be divided in

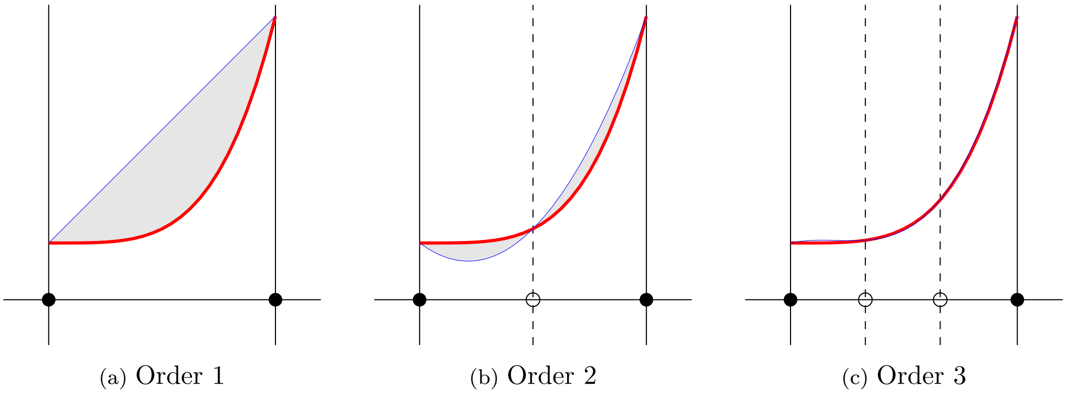

(a) Order 1 (b) Order 2 (c) Order 3

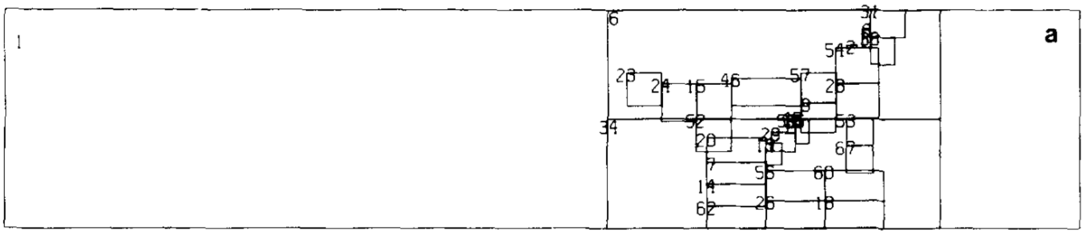
Figure 1.5: Representation of the order of approximation of 1D elements. In this example, the interpolation (blue) of the solution (red) is exact at vertex (black points) and additional nodes (black circles). The interpolation error is represented by the gap between the solution and its interpolation (light gray). This error is reduced when the order of the interpolation increases.

two categories: either an initial group of elements is divided in smaller elements (*cell*-based AMR [25, 26], *patch*-based [27, 28] and *block*-based AMR [29]) or the entire domain is completely re-meshed with a new elements distribution at each adaptation step [30, 31], such as illustrated in figures 1.6a and 1.6b. One currently active research field for h-adaptation is the metric-based anisotropic adaptation [30, 32, 33, 34], which allows the generation of anisotropic elements: this is well-suited to follow the features orientation of flow fields. They have been applied to both re-meshing techniques [12, 30, 33, 35], and cell-based AMR [34, 36]. In this thesis, we investigate the application of this method to quad/octree grid structures within the software Basilisk, as this type of grid structure is inherently efficient for h-adaptation (see chapter 2).

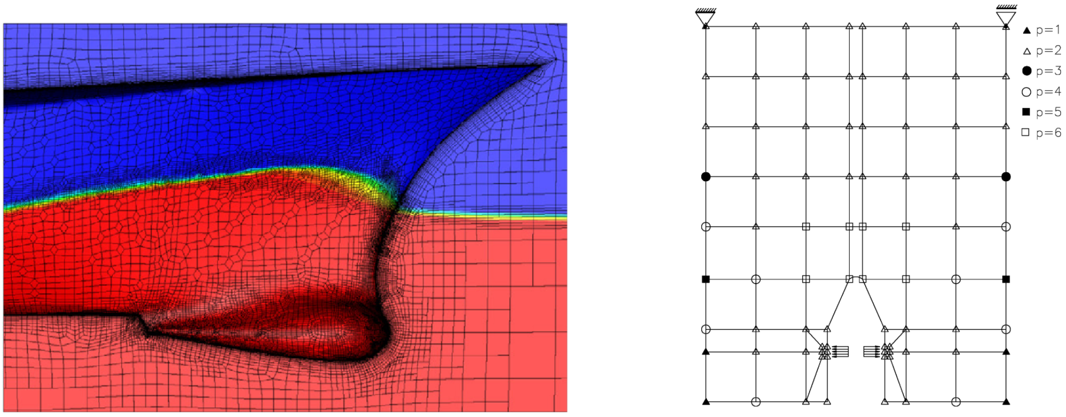
1.2.3 Error estimation in AMR

The role of adaptive methods is to improve the accuracy of a solution in some specified *region of interest*. As the goal of AMR methods is to automatically generate “good” meshes, the first task is to define a suitable criterion that must be automatically computed within the AMR procedure. Many criteria have been defined in order to drive mesh adaptation methods and their efficiency is usually problem dependent. These criteria are generally defined either as an indication of the solution (or its derivatives) variation, or as an estimation of the local error of the solution, if not as an element size prescription field. In this section we recall some classical error criteria.

One first classical approach to define the mesh elements which need to be refined are the *feature-based* estimators. They consists to estimate physically meaningful properties of the flow – the *sensor*, such as gradient detection when dealing with shock waves [38, 39], vortex detection [40, 41] or interface detection in multiphase flows [1, 42, 43]. They were historically based on derivatives (gradients or Hessians) and differences of flow field quantities. These have been applied successfully for a wide range of cases [1, 38, 39, 40, 41, 42, 43], but they require experimented users and special care: these methods perform well at tracking the dominant flow features, but the lack of detection of less dominant structures can lead to the converge toward a wrong solution [39].

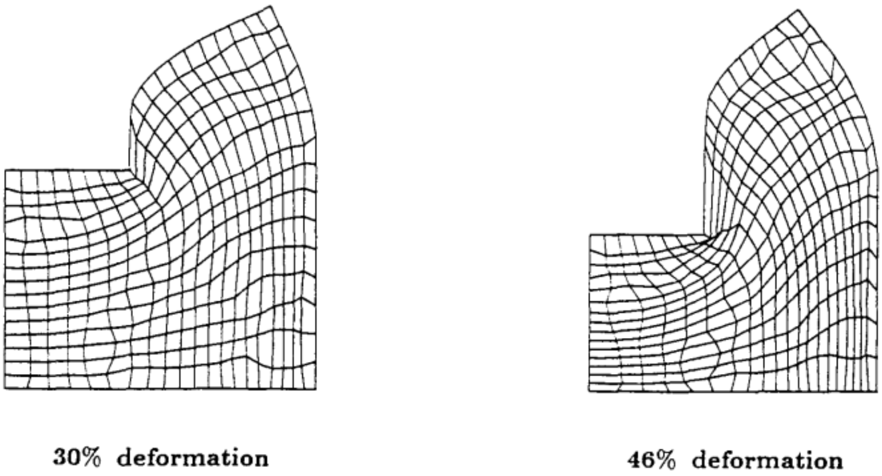


(a) Patch-based adapted mesh. Shock reflection off an oblique wedge [28]: grid hierarchy. Patch 1 is at level 1 (element size $h_1 = 0.02$). Patches 6 and 34 are at level 2 ($h_2 = \frac{h_1}{4}$). Other patches are at level 3 ($h_3 = \frac{h_2}{4}$).



(b) Metric-based adapted mesh. Water surface for the DTMB 5415 hull at $F_r = 0.28$ [34]

(c) p-adapted mesh [37]. A notched concrete plate subjected to two equal and opposite horizontal forces.



(d) r-adapted mesh [23]. Depending on the applied deformation, the mesh elements are deformed.

Figure 1.6: Examples of adapted meshes.

Interpolation error-based estimators [30, 44, 45, 46] are *feature-based* error estimators that try to overcome the default of feature-based estimators. They are based on the numerical solution (velocity $\boldsymbol{u}$, pressure p, ...) but estimate the error on these fields. The interpolation error is defined as the difference between a continuous solution u and its interpolation $\Pi_h u$ on a discrete mesh: $|u - \Pi_h u|$. For the special case of linear interpolation, this error is governed by the second derivatives of a physical field. Among the interpolation error estimates, high-order accurate estimates of the solution [47, 48, 49, 50, 51, 52] are often employed. They are generally based on Richardson extrapolation and are efficient for relatively smooth solutions, but require to compute the solution on a fine mesh and on a coarser mesh, resulting in an error estimation on the coarser mesh.

Another classical interpolation error estimates, the metric-based error estimates [30, 36, 45, 46, 53] are of particular interest. They are based on the concept of Riemannian metric space, which is a continuous representation of a mesh, and allow to compute the interpolation error in various $\boldsymbol{L}^p$-norm: that leads to the possibility to take into account efficiently all the scales or the flow features. This framework is particularly well-suited to obtain anisotropic elements, which have different size in different direction. This is particularly useful when the flow features have privileged orientations. The metric-based refinement method has been developed for tetrahedral and triangular elements and successfully applied in numerous cases [30, 43, 45, 54]. However, the application of this method to hexahedra and quadrangular elements is far less studied. Two main directions are under development to generate these elements using the metric based theory. The most usual is to generate tetrahedral elements, and recombine them to form hexahedra. Recent works propose to generate quasi right-angled triangles and tetrahedra, allowing for easy recombinations into hexahedra. These particular elements are obtained through a point-distribution energy minimization process done by the mesh generator [54, 55, 56]. The second approach is directly based on hex-meshes [34, 36, 43, 53]. In these works, a set of initial quad/hex-elements are refined (or coarsened) based on a metric-based size criterion. This refinement (or coarsening) consists in successive element divisions (or merging).

Nonetheless, *feature-based* (including interpolation error-based) error estimates have a common property: they are only based on the local representation of the numerical solution. Thus, they are independent from the equations being solved, and they don't take into account the errors that can propagate.

In order to consider the influence of the solved equations and their discretization on the solution accuracy, many error estimates have developed. They can be divided into two major class: the *residual-based* error estimators and the *adjoint-based* error estimators (the adjoint-based methods will be shortly described in the next paragraph). The residual-based methods [57] originate mainly from the finite-element mathematical context and rely on the estimation of the residuals: the difference between the discretized equation and its continuous counterpart. Residuals are regularly used in two error estimator families: the *residual-based* errors [58, 59, 60, 61, 62] and the *error transport equation* [63, 64, 65, 66, 67]. In the first case, the residual is directly used to bound the discretization error in energy norm. It depends on the solved equation and their discretization and must be either analytically determined or approximated and reconstructed, which often necessitate to estimate quantities on different (nested) grids. In the second family, a (linearized) equation governing the discretization error is written, and the residual is the source term of this equation. Considering that fact, the region of higher residual amplitude are generally consider as the site of production of the discretization error. This second method intrinsically takes into account the transport of error from regions with insufficient resolution [57, 63], but may lead to over-refinement in the regions where it is transported – referred as *pollution error* [68], instead of the region it is produced, resulting in sub-optimal performances [57, 68, 69]. Furthermore, this require to solve an additional equation or set of equations.

The last approach to obtain adaptive grids originates from industrial purposes. Indeed, one of the industry's main concerns is often to obtain accurate localised integral-based quantities – the *quantities of interest*, such as lift and drag coefficient in aeronautic. In those cases, some parts of the numerical domain are not relevant for the computation of these integral quantities and don't need to be resolved as accurately than the region of interest: resolve more accurately these non relevant regions don't necessarily increase the accuracy of the engineering output of interest [70, 71]. The error estimates which expresses the error in function of these quantities of interest (or output functional) are called *adjoint-based* or *output-based* [72]. They are derived from the equations solved – in particular the variational formulation of the Partial Differential equaitons (PDE) describing the physical problem – and necessitate the resolution of an additional equation: the *adjoint* equation. They generally provides better error rate of convergence regarding the quantity of interest than other error estimations [73]. Different adjoint-based error estimates have been developed based on different underlying theories. The resulting error is often expressed depending on the weighted residual errors of primal and/or dual variables of the problem [72, 73, 74, 75, 76]. However, other forms of adjoint-based error estimates exist, such as the estimates developed in the goal-oriented [30, 32, 33, 35] and the norm oriented framework [77, 78, 79]. These methods are based on the metric-based continuous framework and propose adjoint-weighted interpolation errors. Recent, adjoint-based theoretical developments tend to propose a single adjoint estimation simultaneously representative for two or more quantities of interest [76, 78].

In this thesis, we mainly focus on the metric-based AMR approach applied to octree meshes. In particular, we propose a new extension of the metric-based interpolation errors for quad/octree grids (chapter 3). However, the use of an error estimate to drive mesh adaptation is generally not sufficient by itself, and a special care to the quality of the adapated mesh is essential.

1.3 Mesh quality

Adapted meshes generated directly from a given refinement criterion might present bad quality elements (or metric), and, in case of anisotropic refinement, fast element size transition occurs, such as illustrated in figure 1.7. This is known to result in poorly accurate numerical solutions (see for exemple [80]). In the context of metric-based mesh adaptation, a metric field is provided to the meshing software, which generates the mesh, and either a post-process treatment of the mesh or a pre-process treatment of the provided metric field is needed to improve the mesh elements quality. A part of this treatment is often performed by controlling the cell size variation, and is called *gradation*. This can take many forms, such as, for example, obtain mean local elements sizes from the provided metric field [81], limit the gradient size prescription field and the size ratio between neighbouring elements [2, 80, 82], or solve a minimization problem [83].

The Basilisk quad/octree meshes only contain square or cubic elements obtained by merging or dividing previously existing elements, and is integrated within the the solver: Basilisk does not dispose of a coupling with a meshing software (such as *feflo* [15], *mmg3d* [16] or *gmsh* [17]). Thus, the pre- or post-processing of a mesh is not by default compatible with the code structure. However, the quality of the adaptive remeshing and its impact on the solution accuracy has been investigated and the introduction of a mesh quality (or metric quality) measure has been proposed in this thesis as an additional constrained to our mesh optimisation problem. This method intrinsically limits the ratio between the mean element size of a mesh and its minimal cell size, and we prove it to help reducing the total error of

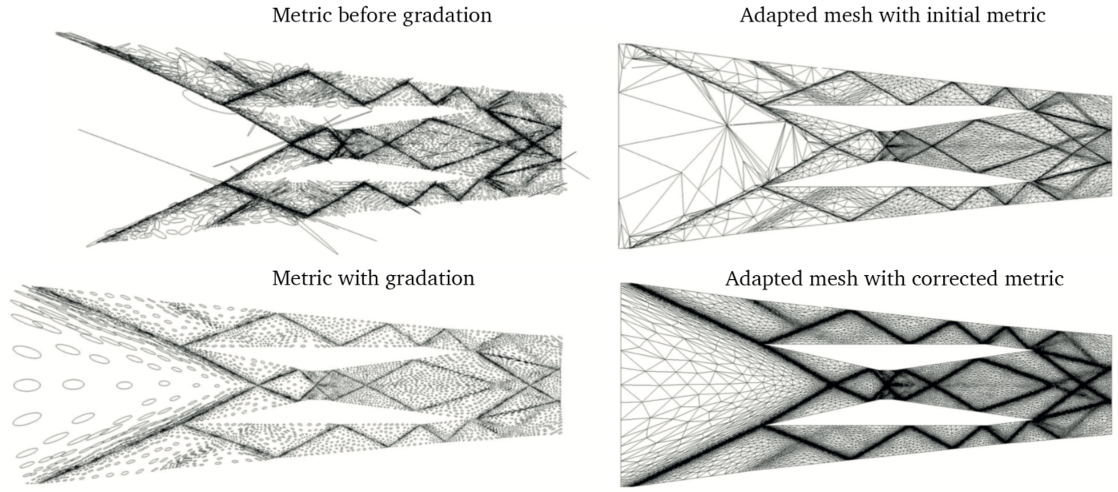
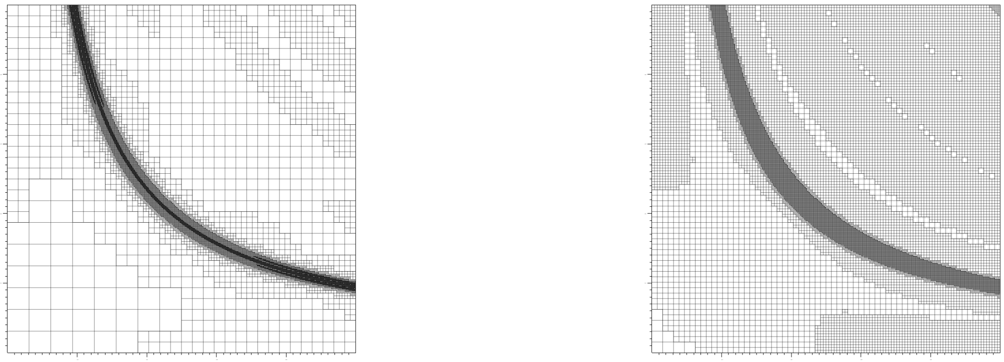


Figure 1.7: Mesh before (bad quality elements) and after (better quality elements) gradation associated with their prescribed metric fields. (images extracted from [80])

some numerical solutions. As a consequence, for a given number of elements, the element size variation throughout the mesh is reduced in comparison with an unconstrained adaptation method, and this without explicitly controlling the gradation (see figure 1.8 for an example). Thus, the proposed method presents similarities in terms of resulting adapted meshes with gradation methods.



(a) Unconstrained mesh.

(b) Constrained mesh.

Figure 1.8: 30 000 elements meshes comparison. Left: the ratio between the minimal and the mean cell size is not constrained. Right: this ratio is constrained. As a consequence, the element size variation is reduced.

2 Content of this thesis

This thesis is organised in 6 chapters, the first one being the introduction. Chapter 2 describes the key components of the software Basilisk used during this thesis. The focus is made on the grid structure, two solvers – Poisson-Helmholtz and incompressible Navier-Stokes, and the AMR method of reference in Basilisk. The incompressible Navier-Stokes solver is a projection-based solver where at the projection step we intrinsically solve a Poisson-Helmholtz equation.

Thus, it relies on the Basilisk multigrid iterative Poisson-Helmholtz solver. The AMR method implemented in Basilisk uses a feature-based error model based on *wavelets* [84], and it belongs to the field of multiresolution analyses. At the end of the chapter, some limits of this AMR method are shown that motivates the need of a new, more robust, AMR approach.

Chapter 3 describes theoretical aspects of metric-based interpolation error estimation. It presents first the continuous mesh framework, which provides the notion of continuous mesh. This continuous mesh is representative for a discrete mesh and developed for simplicial elements. In this framework, an adaptive mesh problem is set as a continuous minimisation problem whose solution is the anisotropic optimal metric. Then, this framework is extended to the case of the quad/octree grids of Basilisk, which means to square and cubic elements. A new isotropic metric-based interpolation error estimate for square/cubic elements is proposed, and it provides an estimation of the minimal interpolation error obtainable on Basilisk meshes. When used as a refinement criterion, this error estimation is shown to provide better adapted meshes than the *wavelet*-based AMR method of Basilisk on analytic functions.

Chapter 4 focuses on numerical errors obtained on regular solutions of the Poisson-Helmholtz equation. It is shown that the adapted meshes obtained by the direct minimization of the interpolation error does not always guarantee the minimization of the numerical error, and it is observed that the size aspect ratio between the finest cell size and the mean cell size of an adapted mesh – the mesh compression – plays a crucial role on the response of the errors of a numerical solution. From that, an error estimation method is proposed assuming that the interpolation error propagates only in the neighbouring mesh elements due to the discretization of the equation, and the classical user-imposed minimum element size criteria set to avoid over-refinement is replaced by an automatic estimation of the minimal element size. This estimation directly depends on an estimation of the continuous solution and the user-defined objective number of grid points. An algorithm based on the quad/octree grid structure and a theoretical metric-based estimation of the mesh compression are proposed and compared. The AMR procedure which results from this chapter shows good performances as illustrated on various examples.

Chapter 5 proposes a generalization of the error estimate for the incompressible Navier-Stokes equations. It relies on a rewritting of the incompressible Navier-Stokes equations. The obtained error estimate is shown to depend on the natural variables (pressure, velocity) of the Navier-Stokes equations. This error estimate is used in combination with the previously defined AMR method and its performances are shown on several test-cases, for various Reynolds numbers and in the limiting cases of the Euler equations. This thesis ends with a conclusion and perspectives on future developments chapter.

2| Basilisk: a software for partial differential equation solutions

Basilisk [6, 85] is an open-source numerical solver dealing with a large range of partial differential equation systems, including the Poisson-Helmholtz equation [25] and the incompressible and the compressible Navier-Stokes equations [86], as well as the Saint-Venant equation [87] among many others [88].

In this chapter we summarize some meaningful ingredients of Basilisk regarding the targeted applications of this thesis. First, Basilisk relies on Cartesian tree-structured grids [26] named quad/octrees, which allows for an efficient coding of numerical schemes.

Then, several Basilisk solvers resort to the solution of a Poisson-Helmholtz equation [41, 86], including a Navier-Stokes incompressible solver which relies on this type of equation in some intermediate step of the numerical procedure. We describe these two solvers and a focus is made on the pure multigrid Poisson-Helmholtz solver [25].

Finally, the quad/octree structure is convenient to perform multiresolution analyses. Such an analysis leads to the *wavelet-based* AMR algorithm [26], which is the AMR method of reference implemented in Basilisk. Some limits of this method are highlighted.

Contents

1 Quad/octree grid structure 14
2 Numerical solvers 16
  2.1 Poisson-Helmholtz solver 17
  2.2 Incompressible Euler/Navier-Stokes solver 18
3 Wavelet-based AMR method 20
4 Limits of the wavelet-based AMR method 22
5 Conclusion 25

1 Quad/octree grid structure

One of the key ingredients in the pre-processing step of numerical simulations is the definition of the mesh discretizing the geometry of interest. This operation is often performed through a mesher: a specific software responsible for mesh generation, such as Gmsh [17], Mmg3d [16], Bamg [89], Yams [90], ... The mesh is then an input for a numerical solver, which solves systems of partial differential equations. With the recognition of the importance of the meshing step in numerical simulations and the need for automated mesh generators ready to use by industrials, many codes have nowadays fully integrated this step as a built-in tool (Star-CCM+ [91], Feflo [15], ...). Basilisk is one of these softwares. It contains its own grid structure – briefly described below – which is a basis for the numerical schemes implemented.

Basilisk grid structure relies on Cartesian *tree*-based structures called quadtrees (in 2D) and octrees (in 3D) [26]. Quad/octree grids are an efficient way to create AMR meshes. They are designed to facilitate cell refinement and coarsening by only allowing to divide or merge cells. The quadtree (*resp.* octree) grid structure is composed with squares (*resp.* cubes). The core concept is that each square (*resp.* cube) can be divided into four squares (*resp.* eight cubes): the cell before division is referred as a *parent* cell, and the four (*resp.* eight) new cells are its *children.* This structural link between *parent* and *child* cells defines the tree-based grids. For the rest of this chapter, we will focus on quadtrees. The generalization to octrees is direct.

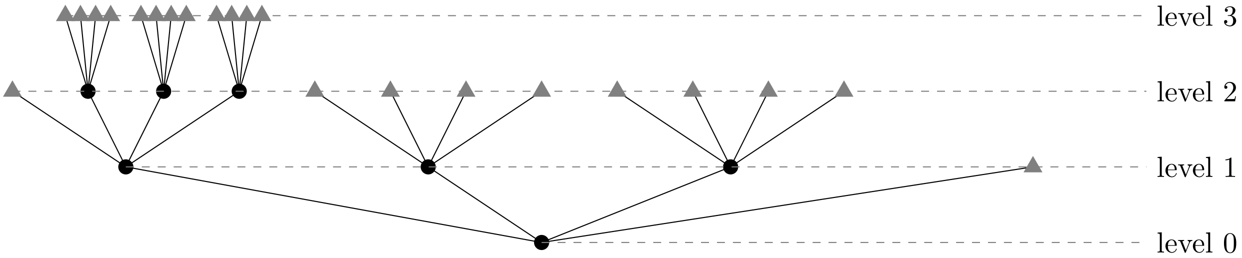
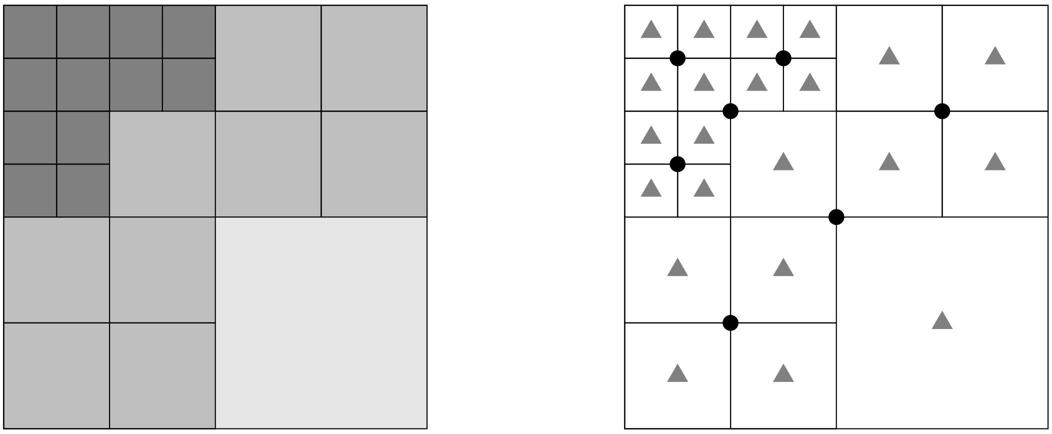


Figure 2.1: A quadtree grid structure corresponding to meshes on figures 2.2 and 2.3. The mesh elements (*ie.* the leaf cells) are represented with gray triangle. Parent cells are represented with black dots. The level 1 cells are the *children* of the level 0 cell. The level 0 cell is the *parent* of the level 1 cells.

Figures 2.1, 2.2 and 2.3 illustrate the principle of quadtree grids. These tree grids are defined from an original element called the *root* cell. The tree which can be constructed through successive cell divisions is represented on the figure 2.1. The cells obtained after a given common number of division of the root cell are by definition at the same hierarchical *level* l. Each cell having the same level l has the same number of parent cell, and the same size h. The cell size is thus directly linked to the *level* : assuming that the size of the root cell representing a square domain is L_0, the size h of a level l cell follows equation (2.1). Thus, the element size distribution is naturally shown through the *level* distribution.

$$h = \frac{L_0}{2^l} \ . \tag{2.1}$$



(a) The mesh elements (*leaf* cells) are colored by levels. From lighter gray to darker gray, the represented levels are 1, 2 and 3.

(b) As in figure 2.1, leaf cell centers are marked with gray triangle. All parent cell centers are marked with black dots.

Figure 2.2: A quadtree mesh corresponding to the quadtree structure represented on figures 2.1 and 2.3.

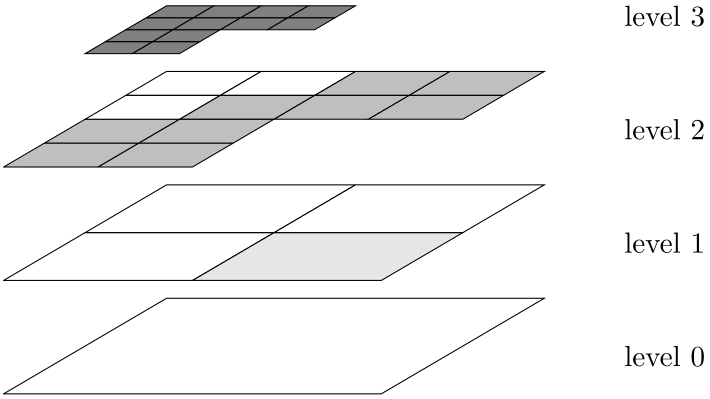


Figure 2.3: A 3D representation of the tree-structure of the mesh illustrated on figures 2.1 and 2.2. Leaf cells are colored by level in gray (from lighter to darker gray, level 1 to 3). Parent cells are not colored. The level 0 cell is the root cell.

Through this structure, a mesh is constructed, as illustrated in figure 2.2a. The tree grid structure contains two types of cells: the cells without children – called *leaf* cells, or *leaves* – and the cells with children – called *parent* cells. The leaves constitute the mesh on which the solution is computed. A restriction operator is defined – generally a volumic mean, but other restrictions may be defined depending on the application –, and the restriction of the solution is stored in the parent cells.

A mesh obtained with Basilisk has some constraints:

- the mesh elements are squares in 2D and cubes in 3D. These are naturally completely isotropic ;
- it follows that the mesh elements orientation is imposed and identical for all cells ;
- the cell size range forms a discrete set of sizes governed by the cell levels l, *cf.* equation (2.1). The level 0 cell is the *root cell*, the common parent of all cells ;

- another constraint is added for practical implementation and numerical scheme stability reasons: the difference between the levels of neighboring cells can't be higher than one, as illustrated in figure 2.4. That means that the size ratio between two neighbors is either equal to 1 or 2. This rule is known as *1-irregularity* in the finite element context [92, 93].

All of these constraints have an impact on the adapted meshes obtained with Basilisk. This will be discussed later in chapter 3.

(a) A mesh respecting Basilisk requirements.

(b) A mesh which does not respect Basilisk requirements.

Figure 2.4: Examples of meshes respecting (or not) the Basilisk requirements. The neighboring cells which do not respect Basilisk mesh constraints are filled in red.

In order to simplify the formulation of numerical methods, and in particular to compute gradients and fluxes related to the the numerical schemes developed in Basilisk, a local regular Cartesian stencil is introduced for each cell as shown in figure 2.5. This stencil is defined for uniformly refined meshes as well as for adapted meshes and uses matrix-like indexation around the main cell of index $[0, 0]$. For uniform meshes, the stencil is composed by neighbouring leaf cells, while for adaptive meshes, it is composed by same size neighbouring leaf cells if they exist, or by *halo* cells. These *halo* cells are ghost cells with the same size than the current cell, and they contain interpolated values of the fields [85].They are naturally defined through the quadtree structure. The matrix-like indexation of this stencil provides a convenient way to represent a cell and its neighbours : in its own referential, a cell has the coordinate [0] in 1D, and its left and right neighbours have respectively the coordinates $[-1]$ and [1]. On the same idea, its left and right faces have the coordinates $[-1/2]$ and $[1/2]$. Figure 2.5 details these local coordinates in 2D, and the generalization to the 3D case is direct.

2 Numerical solvers

Basilisk provides a wide range of numerical solvers to solve various systems of partial differential equations. In the context of this thesis, we focus on a Poisson-Helmholtz solver and an incompressible Navier-Stokes solver. The Poisson-Helmholtz solver is a key component of the incompressible Navier-Stokes solver, as it will be explained in section 2.2. Both solvers use *cell-centered* second-order accurate schemes, and their numerical discretisation is based on the local Cartesian stencil previously defined. In the next section, we will briefly describe their main characteristics.

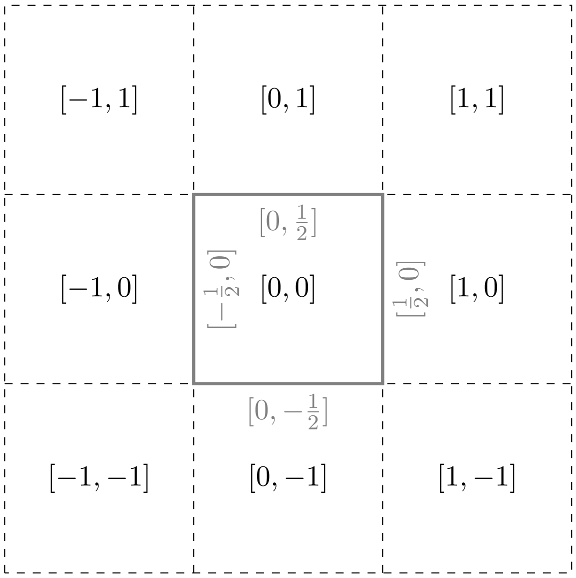


Figure 2.5: Local 3×3 stencil around a cell indexed $[0, 0]$. Face and neighbours matrix-like indices notations are shown. Faces (gray indices) are indexed with $\pm\frac{1}{2}$ and neighbouring cells (black indices) with ± 1. In practice, a 5×5 Cartesian stencil is defined in Basilisk.

2.1 Poisson-Helmholtz solver

Let Ω be a subdomain of $\mathbb{R}^n$, with n the dimension of the problem. For all $\boldsymbol{x}$ in Ω, the Poisson-Helmholtz equation writes

$$\nabla \cdot (D \nabla u) + \lambda u = s \ , \tag{2.2}$$

with $D(\boldsymbol{x})$ a diffusion coefficient, $\lambda(\boldsymbol{x})$ a reaction coefficient, $s(\boldsymbol{x})$ a source term and $u(\boldsymbol{x})$ the solution.

We introduce a discretization of the computational domain Ω_h using the quadtree structure such that $\Omega_h = \bigcup_{i=1}^{N} K_i$, with K_i the i-th element and N the number of elements. We apply the finite volume method to equation (2.2). For each cell K_i, we obtain

$$\int_{\partial K_i} \mathcal{F}(u_i) \cdot \boldsymbol{n}_i \, dS + \int_{K_i} \lambda u_i \, dK = \int_{K_i} s_i \, dK \ , \tag{2.3}$$

where ∂K_i denotes the boundary (the edges) of the element K_i, $\boldsymbol{n}_i$ is the outer normal to the edge, u_i is the mean value of the solution u on K_i, and $\mathcal{F}(u_i) = D \nabla u_i$ is the flux. The volumic integrals are estimated using the midpoint rule. The integral of fluxes is decomposed on cell faces as

$$\int_{\partial K_i} \mathcal{F}(u_i) \cdot \boldsymbol{n}_i \ dS = \sum_{j \ faces} \left(\int_{\partial K_i} (\mathcal{F}(u_i))_j \cdot \boldsymbol{n}_i \ dS \right) \ . \tag{2.4}$$

The fluxes $(\mathcal{F}(u_i))_j \cdot \boldsymbol{n}_i$ are evaluated using face-centered derivatives.

Introducing the element length h and using the matrix-like notations, the 1D discretisation of the Poisson-Helmholtz equation simplifies as

$$\frac{D_{[1/2]} u_{[1]} - (D_{[-1/2]} + D_{[1/2]}) u_{[0]} + D_{[-1/2]} u_{[-1]}}{\Delta^2} + \lambda u_{[0]} = s_{[0]} \ . \tag{2.5}$$

The extension to the 2D and 3D cases is direct.

The discretization (2.5) writes under matrix form as

$$\mathcal{A}\,u = b \; . \tag{2.6}$$

The solution of the linear system (2.6) is the basis of the Poisson-Helmholtz multigrid iterative solver [25, 85]. This type of solver is known to be efficient when dealing with elliptic and parabolic problems, such as the Poisson-Helmholtz equation. In particular, multigrid solvers are asymptotically optimal, as they require $O(N)$ operations to solve the problem 2.2 on a mesh with N grid points [94]. Multigrid solvers are also suitable for the quad/octree grid structure of Basilisk: these methods rely on a multigrid hierarchy which is naturally handled by quadtrees.

Let us introduce $\mathcal{L}$ the linear operator associated to the Poisson-Helmholtz equation. Equation (2.2) writes thus

$$\mathcal{L}(u) = s \; . \tag{2.7}$$

In practice, starting from an initial guess $\tilde{u}$, we solve equation (2.7) for a correction du, such that $u = \tilde{u} + du$. Indeed, du verifies

$$\mathcal{L}(du) = R \; , \tag{2.8}$$

where $R = s - \mathcal{L}(\tilde{u})$ defines the residual associated to equation (2.7). The same form of equation is obtained if we consider the discrete operator $\mathcal{A}$ introduced in equation (2.6) instead of $\mathcal{L}$.

Equation (2.8) is solved and an improved, corrected numerical solution u is obtained from $\tilde{u}$ and du. This process is repeated until the residual R is smaller than a user-defined value of the tolerance of the solver. The equation (2.8) itself is solved using a multigrid solver. The essence of geometric multigrid methods is to decompose the error of the solution on a set of successively coarser grids. This error is then reduced on each grid using relaxation techniques, such as Jacobi or Gauss-Seidel ones. In practice, the multigrid solver of Basilisk operates a multigrid V-cycle. It begins on the grid containing the finest cells, where the initial residual R is computed. This residual is transferred to coarser grids through a volume average restriction. Then, on the coarsest grid, the correction du is computed. After that, this correction is injecting from a level l grid to a level $l+1$ grid through a prolongation operator, and four relaxation steps (described in [25]) are applied to obtain a corrected du. This is repeated from the coarsest grid to the finest grid. The overall procedure is described in the algorithm 1. This solver achieves second order spatial convergence even on complex domains [25].

2.2 Incompressible Euler/Navier-Stokes solver

We give next a brief description of the incompressible Navier-Stokes solver. We look for the solution of the following system of equations:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2.9}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\,\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nabla \cdot (2\nu \boldsymbol{D}) \tag{2.10}$$

Algorithm 1: Multigrid iterative Poisson-Helmholtz solver adapted from [25]

```
Compute the residual $R_L$ on the finest grid $\mathcal{M}_L$;
while $||R_L|| < \varepsilon$ do
    for $l = L_1$ to 0 do
        Compute the residual $R_l$ by volume weighted average of $R_{l+1}$;
    end
    Apply relaxation operator on $\mathcal{M}_0$ to obtain $du$;
    for $l = 1$ to $L$ do
        Get the initial guess $du$ at level $l$ from level $l-1$;
        Apply four times the relaxation operator to update $du$;
    end
    Correct $u = \tilde{u} + du$ on $\mathcal{M}_L$;
    Compute $R_L$ on $\mathcal{M}_L$;
end
```

where $\boldsymbol{u}(\boldsymbol{x})$ holds for the velocity, $p(\boldsymbol{x})$ is the pressure, ρ is the density, ν is the kinematic viscosity, and $\boldsymbol{D}(\boldsymbol{x})$ is the deformation tensor defined as $\boldsymbol{D} = (\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T)/2$. Equation (2.9) reflects the conservation of mass and equation (2.10) is the momentum equation.

The incompressible Navier-Stokes equations are typically identified with the system of equation (2.9) and 2.10 which can be discretized in time between t^n, the current time, and t^{n+1}, the next time after integration of the equation during a time step δt. For single phase flows and imposing the incompressibility condition, the system of equations above can be written as a coupled system of Helmholtz-Poisson equations for the different components of the velocity u_i and the pressure

$$\nabla \cdot ((\nu\delta t)\nabla u_i) - u_i^{n+1} = -u_i^* + \frac{\delta t}{\rho}\nabla p \tag{2.11}$$

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p\right) = \frac{1}{\delta t}\nabla \cdot \boldsymbol{u}^* \tag{2.12}$$

where we have introduced the definition of the auxiliary velocity field

$$\boldsymbol{u}^* = \boldsymbol{u}^n - \delta t\,(\boldsymbol{u} \cdot \nabla)\,\boldsymbol{u}\ , \tag{2.13}$$

In Basilisk, the set of above equations are solved using a classical time-splitting projection method [95, 96, 97, 98, 99, 100] through the following steps [25, 101, 102]: an *advection* step, a *diffusion* step and a *projection* step.

The **advection step** computes $\boldsymbol{u}^*$ by rewriting equation (2.13) using the incompressibility constraint:

$$\boldsymbol{u}^* = \boldsymbol{u}^n - \delta t\,\nabla \cdot (\boldsymbol{u}\boldsymbol{u})\ . \tag{2.14}$$

The divergence term is written as a sum of fluxes other the faces of the cell by the mean of the Green-Ostrogradski theorem. For each component of the velocity u_i, we obtain

$$u_i^* = u_i^n - \frac{\delta t}{h}\sum_{faces\ f}(\boldsymbol{u}_f \cdot \boldsymbol{n}_f\, u_i^n)\ , \tag{2.15}$$

with h the cell size and $\boldsymbol{n}_f$ the outward normal to the face f. The face velocity $\boldsymbol{u}_f$ is obtained using the Bell-Colella-Glaz scheme [25, 103]. It proceeds in two steps: first, the face velocity

$\boldsymbol{u}_f = \boldsymbol{u}(\boldsymbol{x} \pm h/2)$ at time $t + \frac{\delta t}{2}$ is expressed from the cell centered velocity through a Taylor expansion:

$$u_i^{n+1/2}(\boldsymbol{x} + h/2) = u_i^n(\boldsymbol{x}) + \frac{\partial u_i}{\partial x}\frac{h}{2} + \frac{\partial u_i}{\partial t}\frac{\delta t}{2} + O(h^2, \delta t^2)\ . \tag{2.16}$$

Then, the momentum equation is introduced to express the temporal derivative as a spatial derivative. It results in

$$u_i^{n+1/2}(\boldsymbol{x} + h/2) = u_i^n(\boldsymbol{x}) + \frac{\partial u_i}{\partial x}\frac{h}{2} + \frac{\delta t}{2}\left(-\frac{1}{\rho}\nabla p - (\boldsymbol{u} \cdot \nabla)u_i^n\right) + O(h^2, \delta t^2)\ . \tag{2.17}$$

The **diffusion step** solves the Helmohltz-Poisson like equation (2.11) to obtain an estimation of the velocity field $\boldsymbol{u}_{est}^{n+1}$ at the end of the timestep using the pressure gradient evaluated at t^n.

The resulting velocity field $\boldsymbol{u}_{est}^{n+1}$ needs to be corrected as it does not satisfy the incompressibility condition. This is achieved during the **projection step** in which pressure is obtained at time $t^n + \delta t$ by solving the Poisson equation (2.12) using the previously described multigrid iterative Poisson solver and computing the source from the temporary vector field

$$\boldsymbol{u}^{**} = \boldsymbol{u}_{est}^{n+1} + \frac{\delta t}{\rho}\nabla p^n\ . \tag{2.18}$$

which satisfies

$$\nabla \cdot \boldsymbol{u}^{*} = \nabla \cdot \boldsymbol{u}^{**}$$

The final divergence free velocity field is readily obtained using the updated pressure field

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{**} - \frac{\delta t}{\rho}\nabla p^{n+1}$$

which can be also written as

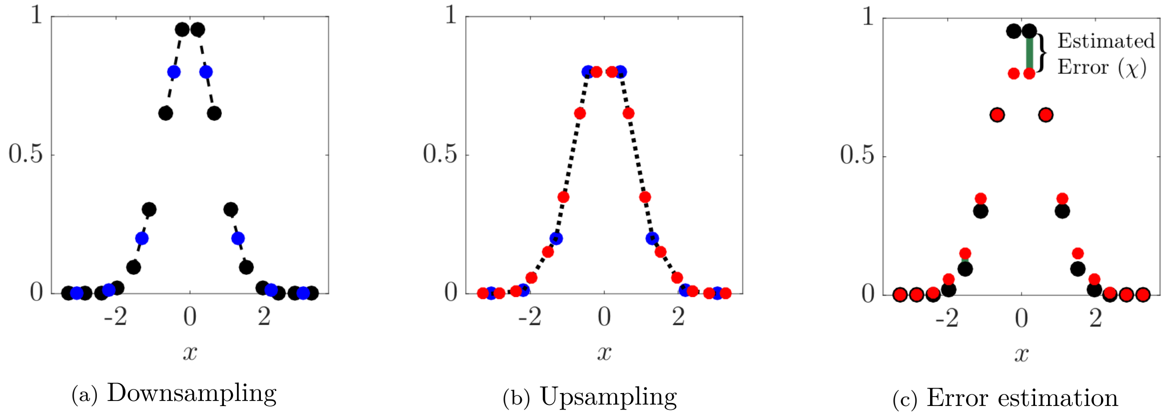
$$\boldsymbol{u}^{n+1} = \boldsymbol{u}_{est}^{n+1} + \frac{\delta t}{\rho}\nabla p^{n} - \frac{\delta t}{\rho}\nabla p^{n+1}$$

3 Wavelet-based AMR method

In order to perform adaptive mesh simulations, Basilisk provides its own AMR method. In this section, we shortly introduce it. It will be used as a reference to discuss the performances of the new AMR method that we propose later in this thesis.

The Basilisk quadtree grid structure is a succession of nested grids. Therefore, it can be trivially used as basis to perform adaptive mesh refinement (AMR) simulations, conditioned by the introduction of an accurate error estimator. Through the succession of nested grids, Basilisk provides access to the data stored in all cells at all levels. In other words, it provides access to the information contained at different spatial resolution. Thus, it is well suited to perform multiresolution analyses [84, 104, 105, 106]. This type of analyses provides errors estimators [107, 108, 109] based on wavelets functions. The initial AMR method implemented in Basilisk – called *wavelet*-based AMR [26] – is based on one of such estimator. Due to the mathematical properties of these functions, mesh adaptation procedures led by *wavelet*-based error estimators naturally produce larger elements in regions with smooth data than more classical mesh adaptation methods [84]. This increases the data compression compared to other adaptation methods.

The wavelet-based error estimation in Basilisk is decomposed in three steps and involves a parent cell of level $l-1$ and its children of level l. An illustration is depicted in figure 2.6.



(a) Downsampling (b) Upsampling (c) Error estimation

Figure 2.6: Principle of the *wavelet-based* AMR method (figure adapted from [26]). (a) The field $u(\boldsymbol{x}_c^l)$ (black dots) is downsampled to $u(\boldsymbol{x}_c^{l-1})$ (blue dots), which is then upsampled (b) by linear interpolation to $\left(\Pi u^{l-1}\right)(\boldsymbol{x}_c^l)$ (red dots). (c) The error is estimated as $|u(\boldsymbol{x}_c^l) - \left(\Pi u^{l-1}\right)(\boldsymbol{x}_c^l)|$.

First, we suppose that a field $u(\boldsymbol{x}_c^l)$ is known at the center of the child cells $\boldsymbol{x}_c^l$. This field is first downsampled : $u(\boldsymbol{x}_c^{l-1})$ is estimated at the center of the parent cell $\boldsymbol{x}_c^{l-1}$. This downsample operation is referred as a *restriction* operation. The *restriction* operator is generally a volumic mean of the child cells values. This formulation is exact for finite volume formulation, in which grid cell values represent volume-averaged quantities.

The downsampled field $u(\boldsymbol{x}_c^{l-1})$ must be upsampled to be compared with the original field $u(\boldsymbol{x}_c^l)$. This step is done through a *prolongation* operation. The *prolongation* operation is a second order bilinear interpolation. This defines the interpolation of the solution at the level $l-1$: $\left(\Pi u^{l-1}\right)(\boldsymbol{x}_c^l)$.

Finally, the error is estimated as the difference between $u(\boldsymbol{x}_c^l)$ and $\left(\Pi u^{l-1}\right)(\boldsymbol{x}_c^l)$. In [110], it is demonstrated that this recovers an interpolation error. Below, we explicitly write the second order terms in the Taylor expansions to express this error.

Let consider the Taylor expansion of $u(\boldsymbol{x})$ around the child cell center $\boldsymbol{x}_c^l$

$$u(\boldsymbol{x}) = u(\boldsymbol{x}_c^l) + (\boldsymbol{x}-\boldsymbol{x}_c^l)\,(\nabla u)_{\boldsymbol{x}_c^l} + (\boldsymbol{x}-\boldsymbol{x}_c^l)^T\, H_{\boldsymbol{x}_c^l}(u)\,(\boldsymbol{x}-\boldsymbol{x}_c^l) + O\left((\boldsymbol{x}-\boldsymbol{x}_c^l)^3\right) \;, \tag{2.19}$$

and the Taylor expansion of $u(\boldsymbol{x})$ around the parent cell center $\boldsymbol{x}_c^{l-1}$

$$u(\boldsymbol{x}) = u(\boldsymbol{x}_c^{l-1}) + (\boldsymbol{x}-\boldsymbol{x}_c^{l-1})\,(\nabla u)_{\boldsymbol{x}_c^{l-1}} + (\boldsymbol{x}-\boldsymbol{x}_c^{l-1})^T\, H_{\boldsymbol{x}_c^{l-1}}(u)\,(\boldsymbol{x}-\boldsymbol{x}_c^{l-1}) + O\left((\boldsymbol{x}-\boldsymbol{x}_c^{l-1})^3\right) \;. \tag{2.20}$$

Subtracting equation (2.20) to equation (2.19) at leaf cell center $\boldsymbol{x} = \boldsymbol{x}_c^l$, we readily obtain

$$\begin{aligned} 0 = u(\boldsymbol{x}_c^l) - \Big(u(\boldsymbol{x}_c^{l-1}) + (\boldsymbol{x}_c^l - \boldsymbol{x}_c^{l-1})\,(\nabla u)_{\boldsymbol{x}_c^{l-1}} + (\boldsymbol{x}_c^l - \boldsymbol{x}_c^{l-1})^T\, H_{\boldsymbol{x}_c^{l-1}}(u)\,(\boldsymbol{x}_c^l - \boldsymbol{x}_c^{l-1})\Big) \\ + O\left((\boldsymbol{x}_c^l - \boldsymbol{x}_c^{l-1})^3\right) \ . \end{aligned} \tag{2.21}$$

Recognizing the interpolation of the solution at the $l-1$ level $\left(\Pi u^{l-1}\right)(\boldsymbol{x}_c^l) = u(\boldsymbol{x}_c^{l-1}) + (\boldsymbol{x}_c^l - \boldsymbol{x}_c^{l-1})\,(\nabla u)_{\boldsymbol{x}_c^{l-1}}$, equation (2.21) rewrites

$$|u(\boldsymbol{x}_c^l) - \left(\Pi u^{l-1}\right)(\boldsymbol{x}_c^l)| = |(\boldsymbol{x}_c^l - \boldsymbol{x}_c^{l-1})^T\, H_{\boldsymbol{x}_c^{l-1}}(u)\,(\boldsymbol{x}_c^l - \boldsymbol{x}_c^{l-1}) + O\left((\boldsymbol{x}_c^l - \boldsymbol{x}_c^{l-1})^3\right)| \ . \tag{2.22}$$

This error is an estimation of the interpolation error of u. It is used as a refinement criterion in the AMR method presented in the algorithm 2. The local error is compared with a user-defined ε criterion. If the error is higher than ε, the cell must be refined. If the error is lower than $\frac{2\varepsilon}{3}$, the cell can be coarsen. From an initial mesh, this procedure allows locally one level of refinement or coarsening. It is then repeated at each time step to obtain adapted meshes.

Algorithm 2: Algorithm of the wavelet-based AMR method in Basilisk

```
Set the initial uniform (Cartesian) mesh $\mathcal{M}_0$;
Set the error threshold $\varepsilon$;
Set the maximum authorized level $l_{max}$;
while $t < t_{end}$ do
    Compute the numerical solution $u(\boldsymbol{x}_c^l)$ on $\mathcal{M}_t$;
    foreach cell do
        Compute $u(\boldsymbol{x}_c^{l-1})$ from $u(\boldsymbol{x}_c^l)$ through the restriction operator;
        Compute $\left(\Pi u^{l-1}\right)(\boldsymbol{x}_c^l)$ from $u(\boldsymbol{x}_c^{l-1})$ through the prolongation operator;
        Estimate the error $\chi = |u(\boldsymbol{x}_c^l) - \left(\Pi u^{l-1}\right)(\boldsymbol{x}_c^l)|$;
        if $\chi > \varepsilon$ and $l \leq l_{max}$ then
            Refine the cell;
        else
            if $\chi < \frac{2\varepsilon}{3}$ and neighboring cells are not too fine then
                Coarsen the cell;
            end
        end
    end
end
```

To ensure the neighbouring cell size Basilisk constraint defined in section 1, priority is given to cell refinement over cell coarsening. That means that if a cell twice bigger than one of its neighbours should be coarsen, but not this neighbour, then the cell is not coarsen. A minimal (optional) and a maximum (mandatory) level are imposed by the user to ensure numerical scheme consistency and avoid over/under-refinement. Finally, the choice of the *sensor* is the responsibility of the user.

4 Limits of the wavelet-based AMR method

Although the wavelet-based AMR method of Basilisk has been verified and validated on numerous test cases [6, 25, 26, 87, 111, 112], this method has some limits. First of all, it is today commonly admitted that the best practice to control the interpolation error of a sensor is to evaluate it in $\boldsymbol{L}^2$-norm [12, 30, 113]. For specific cases, the benefits of controlling this error in $\boldsymbol{L}^4$-norm over $\boldsymbol{L}^2$-norm have also been demonstrated [12, 114]. The Basilisk wavelet-based interpolation error estimation doesn't allow to control the error norm, and is not an $\boldsymbol{L}^2$-norm error. Secondly, it is possible to find cases where the Basilisk AMR method seems to perform poorly, as we will see in this section. Thus, the introduction of new error estimators in Basilisk, and in particular an $\boldsymbol{L}^p$-norm interpolation error (chapter 3), should be beneficial.

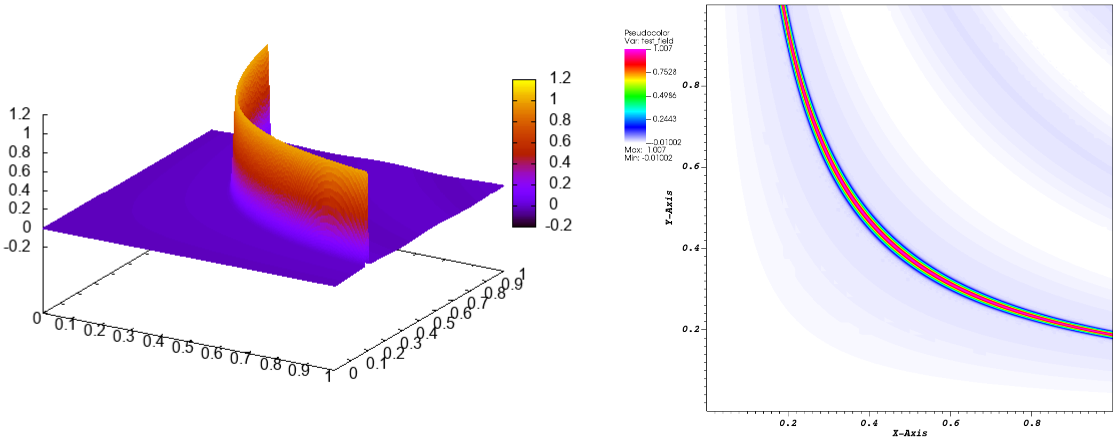


Figure 2.7: Representation of the function $u_1(x, y)$.

We consider the Poisson-Helmholtz equation

$$\nabla \cdot (\nabla u) - u = s \ , \tag{2.23}$$

where u is the solution and s is a source term. The source term s of this equation is analytically computed such that the multiscale sinusoidal solution u_1:

$$u_1(x, y) = \sin(a\,(xy - b))\left(0.01 + e^{-(d\,(xy-c))^2}\right) \ , \tag{2.24}$$

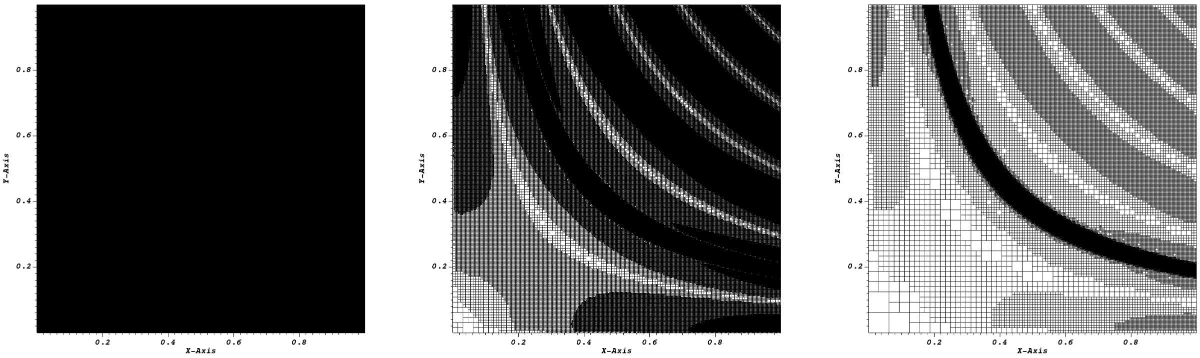
with $a = 5\pi$, $b = 1.5\pi/50$, $c = 3\pi/50$ and $d = 50\pi$. This solution is represented in figure 2.7. It combines an harmonic function with a decaying exponential function. This type of combination is typically encountered in the classical solutions of systems containing instabilities[1].

An $\boldsymbol{L}^1$-norm error convergence study is made on a set of uniformly refined meshes, and it is compared with the error measured on adapted meshes. The uniform mesh levels range from 8 to 11. The adapted meshes have their maximum level fixed to 11, and are obtained by coarsening a level 11 uniform mesh in the region where the error is the smallest, such as represented on figure 2.8. The sensor used for this study is the numerical solution.

The resulting error convergence curve is presented on figure 2.9. The error convergence for the set of uniformly refined meshes – red dots and line – converges at order 2, which is expected for the solver used. The green triangles represent the error obtained on the adapted meshes. Two main observations can be made from these curves:

---

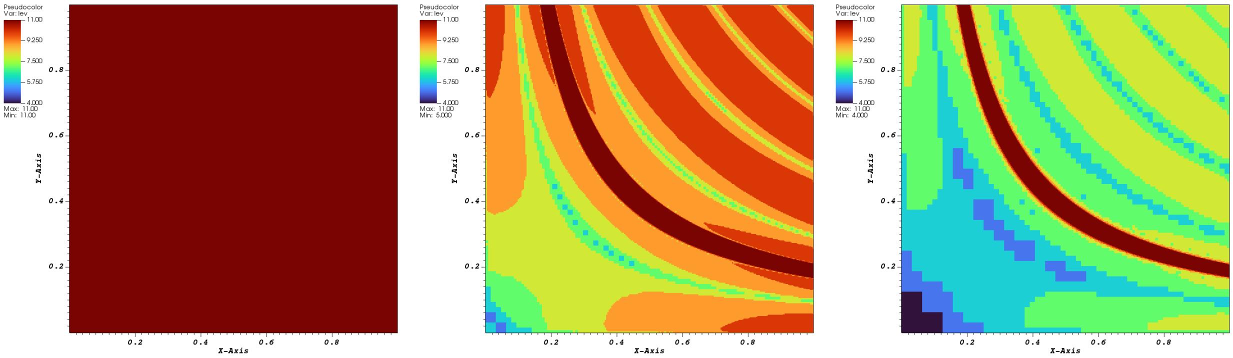
[1] Of course, equation (2.23) is steady whereas unstable flows are not steady, but adaptive simulations are also used for unsteady problems.



(a) Basilisk adapted mesh – point 1 on figure 2.9

(b) Basilisk adapted mesh – point 2 on figure 2.9

(c) Basilisk adapted mesh – point 3 on figure 2.9



(d) Mesh levels – point 1 on figure 2.9

(e) Mesh levels – point 2 on figure 2.9

(f) Mesh levels – point 3 on figure 2.9

Figure 2.8: Mesh and level field obtained for three adaptive criterion thresholds. From left to right, the mesh is coarsen.

1. The error on adapted meshes is at best equal to the error on uniform meshes. That means that, in this case, adapted meshes don't provide any gain in performance compared to uniform meshes.

2. When decreasing number of elements, the error on adapted meshes increases much faster than order 2. That implies that for lower number of elements, the wavelet-based adaptation method is detrimental to the solution accuracy.

This case clearly shows that the wavelet-based AMR method is not always a correct method to reduce the error of numerical solutions. The major issue is that, for a real test-case – with unknown analytical solution, it is not possible to know in advance if the method will be efficient or detrimental.

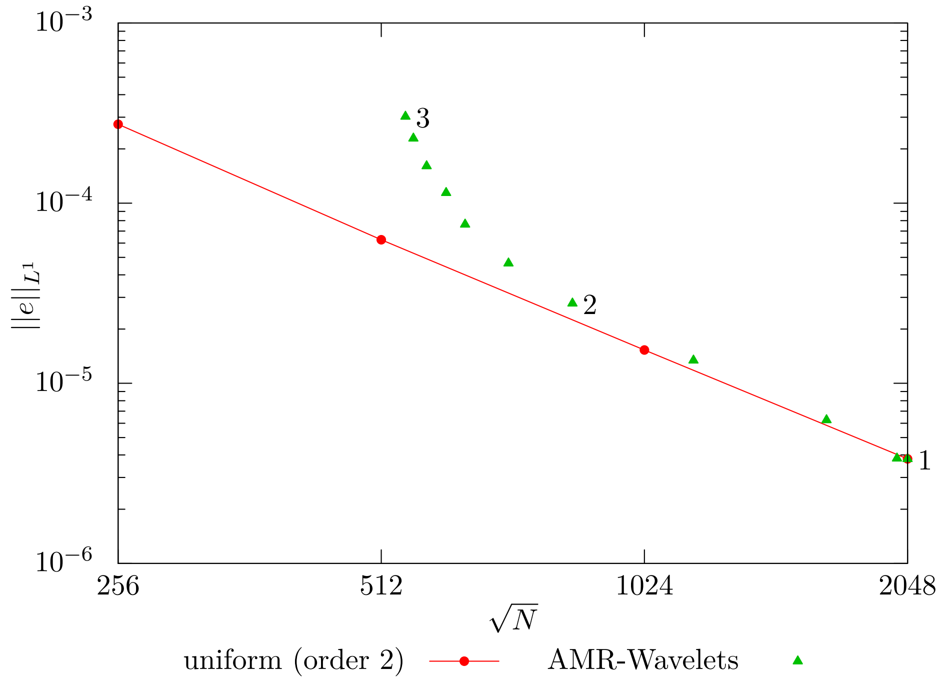


Figure 2.9: $\boldsymbol{L}^1$-norm error *vs.* square root of number of cells. The error is measured on the numerical solution of the partial differential equation (2.23). Red dots and line represent the error convergence on a sequence of uniformly refined meshes. Green triangles represent the error convergence on adapted grids with minimum level $l = 11$. The numbers correspond to the meshes represented on figure 2.8

5 Conclusion

Some key components of the software Basilisk have been described. This software combines both the possibility to create quad/octree meshes and to solve partial differential equations on these meshes. The grid structure is a tree-grid structure, which is known to be an efficient structures to create meshes and perform AMR simulations. Moreover, the Basilisk quad/octree structure offers an efficient way to code numerical schemes through the access to a local Cartesian stencil around each mesh element. However, this structure imposes some constraints on the meshes. In particular, they are isotropic.

Two cell-centered numerical solvers have been presented and will be used in the context of this thesis: a second order multigrid iterative Poisson-Helmholtz solver and a second order incompressible Navier-Stokes solver based on a classical time-splitting projection method, in which a system of Helmholtz-Poisson like equations are solved for the different components of the velocity field and pressure.

Finally, Basilisk provides a wavelet-based AMR procedure which is suitable for its quad/octree grid. This method belongs to the field of multiresolution analysis, and estimates the interpolation error of a sensor. It has been validated on numerous examples with the different solvers implemented within Basilisk. However, some cases show a detrimental behaviour of this AMR method with respect to the solution accuracy of a numerical solver. Moreover, it does not allow to choose a norm for the error computation. This motivates the development of the new AMR that will be presented in the next chapters.

3| Interpolation error model

In the previous chapter, we described the core component of the software Basilisk. The performance of its wavelet-based AMR method has been assessed, and its limits – weak performance for some numerical test-cases, no choice of the error norm – have been emphasized. This motivates the necessity of a new adaptive criterion.

Over the last decades, metric-based adaptive methods have proved their efficiency and robustness mostly for anisotropic unstructured mesh refinement. Based on a continuous representation of the mesh, they provide an estimation of the optimal (minimal) interpolation error for a sensor (feature-based approach) or a quantity of interest (goal-oriented approach), and the corresponding optimal mesh. We investigate the use of Riemannian metric-based adaptive methods applied to the quadtree structure of Basilisk. We present first a short overview of the continuous mesh framework and the metric-based interpolation error estimator [30, 46]. We then particularize the approach for Basilisk quad/octree grids and explicitly provide the optimal (minimal) interpolation error. At the end of this chapter, we validate the Riemannian metric AMR approach on a few analytical examples.

Contents

1 Riemannian metric space for mesh adaptation 28
  1.1 Euclidean and Riemannian metric space 28
  1.2 Unit mesh 30
2 The continuous mesh framework 31
  2.1 Duality discrete-continuous mesh 31
  2.2 Continuous linear interpolate and continuous interpolation error . . 33
3 Continuous interpolation error for quad/octree grids 35
  3.1 Quadtree continuous interpolation error 35
  3.2 Octree continuous interpolation error 38
  3.3 Uniformly refined mesh interpolation error 39
4 Optimal metric-based mesh adaptation 41
  4.1 $\mathbf{L^p}$-norm anisotropic metric-based interpolation error 41
  4.2 $\mathbf{L^p}$-norm isotropic interpolation error for quad/octree grids 43
  4.3 Adaptive Algorithm 45
5 Quad/Octree optimal error validation 46
  5.1 Multiscale example 46

5.2 Boundary layer example 49
6 Conclusion 50

1 Riemannian metric space for mesh adaptation

The introduction of the metric-based mesh generation by George in 1991 [115], who proposed to compute distances in Riemannian metric spaces, has motivated the work of many researchers that have introduced new advances in this field [116, 46, 30, 117, 118, 79]. It is today a mature field of research which has been successfully applied to numerous flow fields, even in the presence of discontinuities. It is particularly applied in the context of anisotropic mesh adaptation as the directional distances provided by Riemannian metric spaces are well suited to create anisotropic elements [30, 53]. Riemannian metric spaces also provide a powerful mathematical tool to apply classical optimization techniques, and particularly to minimize the numerical error in simulations, through the continuous mesh framework, which allows to use continuous mathematical tools such as calculus of variation and differentiation to define and solve such problems. It is possible to prove equivalences between this continuous space and the space representing the discrete meshes. In this section we will recall some fundamentals on Riemannian metric spaces and its link with mesh generation, before moving onto the introduction of the continuous mesh framework in the next section. These sections are only meant to provide a brief overview of the mathematical principles, we refer to [119, 120, 121] for a more detailed description of the methods.

1.1 Euclidean and Riemannian metric space

Euclidean metric space

An Euclidean metric space $(\mathbb{R}^n, (\,\cdot\,,\,\cdot\,))$ is a vector space associated with the definition of a scalar product $(\,\cdot\,,\,\cdot\,)$. This scalar product is by definition a symmetric positive definite form which can be represented by its matrix $\mathcal{M}$ as

$$\begin{aligned} (\,\cdot\,,\,\cdot\,)_{\mathcal{M}} : \mathbb{R}^n \times \mathbb{R}^n &\to \mathbb{R}^+ \\ (\boldsymbol{u}, \boldsymbol{v}) &\mapsto (\boldsymbol{u}, \boldsymbol{v})_{\mathcal{M}} = \boldsymbol{u}^T \mathcal{M} \boldsymbol{v} \ , \end{aligned} \tag{3.1}$$

where $\mathcal{M}$ is frequently called a *metric tensor* or simply a *metric*. The most classical Euclidean space is the canonical Euclidean space. It refers to the physical space and is defined with the metric $\mathcal{M} = I_n$ where I_n is the identity matrix of the space $\mathbb{R}^n$.

From the scalar product, we define the Euclidian norm $||\,\cdot\,||_{\mathcal{M}}$

$$\begin{aligned} ||\,\cdot\,||_{\mathcal{M}} : \mathbb{R}^n &\to \mathbb{R}^+ \\ \boldsymbol{u} &\mapsto ||\boldsymbol{u}||_{\mathcal{M}} = \sqrt{\boldsymbol{u}^T \mathcal{M} \boldsymbol{u}} \end{aligned} \tag{3.2}$$

and the associated distance $d_{\mathcal{M}}$

$$\begin{aligned} d_{\mathcal{M}} : \mathbb{R}^n \times \mathbb{R}^n &\to \mathbb{R}^+ \\ (\boldsymbol{u}, \boldsymbol{v}) &\mapsto d_{\mathcal{M}}(\boldsymbol{u}, \boldsymbol{v}) = ||\boldsymbol{u} - \boldsymbol{v}||_{\mathcal{M}} \ . \end{aligned} \tag{3.3}$$

The scalar product, the norm and the distance allow to define some usual geometrical quantities, and in particular the length of an edge, the angle between two vectors and the volume of an element. The length $l_{\mathcal{M}}$ of an edge $\boldsymbol{e}$ is given by

$$l_{\mathcal{M}}(\boldsymbol{e}) = \sqrt{\boldsymbol{e}^T\mathcal{M}\boldsymbol{e}}\ . \tag{3.4}$$

The angle between two vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ is the unique real number $\theta_{\mathcal{M}} \in [0,\pi]$ which verifies

$$\cos(\theta_{\mathcal{M}}) = \frac{(\boldsymbol{v}_1,\boldsymbol{v}_2)_{\mathcal{M}}}{||\boldsymbol{v}_1||_{\mathcal{M}}||\boldsymbol{v}_2||_{\mathcal{M}}}\ . \tag{3.5}$$

The volume $|K|_{\mathcal{M}}$ of an element K is deduced from the volume of the element in the canonical space $|K|_{I_n}$:

$$|K|_{\mathcal{M}} = \sqrt{det(\mathcal{M})}|K|_{I_n}\ . \tag{3.6}$$

Since $\mathcal{M}$ is symmetric positive definite, it is diagonalizable and admits the decomposition

$$\mathcal{M} = \mathcal{R}\Lambda\mathcal{R}^T\ , \tag{3.7}$$

with Λ a diagonal matrix composed with the eigenvalues $(\lambda_i)_{i\in[\![1,n]\!]}$ of $\mathcal{M}$, and $\mathcal{R}$ an orthonormal matrix composed with the eigenvectors $(\boldsymbol{v}_i)_{i\in[\![1,n]\!]}$ of $\mathcal{M}$.

The geometrical representation (*cf.* figure 3.1 left) of a metric $\mathcal{M}$ is obtained considering the unit ball $\mathcal{B}_{\mathcal{M}}$ associated with $\mathcal{M}$:

$$\mathcal{B}_{\mathcal{M}} = \{x \text{ in } \mathbb{R}^n|\ ||x||_{\mathcal{M}} = 1\}\ . \tag{3.8}$$

It is an ellipse in $\mathbb{R}^2$, and an ellipsoid in $\mathbb{R}^3$. The metric $\mathcal{M}$ provides an application that maps the unit ball associated with I_n to the unit ball associated with $\mathcal{M}$. This transformation is called the *natural mapping* and corresponds to the change of basis from the canonical basis of $(\mathbb{R}^n, I_n)$ to the basis of $(\mathbb{R}^n,\mathcal{M})$. It is illustrated in figure 3.1 (right), and it writes

$$\mathcal{M}^{-\frac{1}{2}} = \mathcal{R}\Lambda^{-\frac{1}{2}}\mathcal{R}^T\ . \tag{3.9}$$

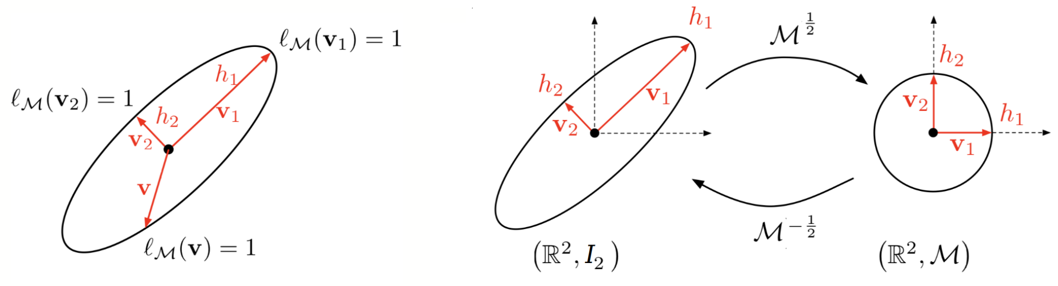


Figure 3.1: Left: a 2D unit ball associated with a metric $\mathcal{M}$. Right: natural mapping between the unit balls of a metric space $(\mathbb{R}^n,\mathcal{M})$ and the physical space $(\mathbb{R}^n, I_n)$. $h_i = \lambda_i^{-\frac{1}{2}}$ are the eigenvalues of $\mathcal{M}^{-1/2}$.

Riemannian metric space

A Riemannian metric space associated to a domain Ω is defined as a continuously varying metric tensor field $\boldsymbol{M} = (\mathcal{M}(\boldsymbol{x}))_{\boldsymbol{x}\in\Omega}$, whereas the metric of an Euclidean space is constant in the whole domain. A complementary point of view on Riemannian metric spaces is that they consist in a set of Euclidian metric spaces: locally, the plane tangent to a Riemannian metric space defines an Euclidean space. Thus, if a global definition of the scalar product and the norm are not defined in Riemannian metric spaces, the geometrical quantities can be extended from local considerations. The measure of a length in a Riemannian space – the shortest distance between two points – is (generally) no longer a straight line, and an integral formulation (geodesic) must be used. Let parametrize the path of a segment $\boldsymbol{AB}$ by $\boldsymbol{\gamma} : t \in [0,1] \mapsto \boldsymbol{A} + t\,\boldsymbol{AB}$. Its length is then defined as

$$l_{\mathcal{M}}(\boldsymbol{AB}) = \int_0^1 ||\boldsymbol{\gamma}'(t)||_{\mathcal{M}(t)}\, dt \tag{3.10}$$

$$= \int_0^1 \sqrt{\boldsymbol{AB}^T \mathcal{M}(\boldsymbol{AB} + t\,\boldsymbol{AB})\boldsymbol{AB}}\, dt \tag{3.11}$$

The volume of a subset $K \subset \Omega$ expresses as

$$|K|_{\mathcal{M}} = \int_K \sqrt{det(\mathcal{M}(\boldsymbol{x}))}\ \mathrm{d}\boldsymbol{x}\ , \tag{3.12}$$

and the angle between two vectors $\boldsymbol{v}_1 = \boldsymbol{AB}$ and $\boldsymbol{v}_2 = \boldsymbol{AC}$ is the real number $\theta_{\mathcal{M}} \in [0,\pi]$ which verifies

$$\cos(\theta_{\mathcal{M}}) = \frac{(\boldsymbol{v}_1, \boldsymbol{v}_2)_{\mathcal{M}(A)}}{||\boldsymbol{v}_1||_{\mathcal{M}(A)}||\boldsymbol{v}_2||_{\mathcal{M}(A)}}\ . \tag{3.13}$$

1.2 Unit mesh

As seen in the previous paragraph, a Riemannian metric space provides local geometrical information. This information includes length, volume and orientation. Riemannian metric spaces are thus well suited to create anisotropic meshes composed with elements constrained by these geometrical quantities. Metric-based mesh adaptation relies on this concept. In particular, since the work of P. George *et al.* in [115], the metric-based meshing softwares (like *feflo* [30] or *mmg3d* [16]) directly compute distances in any given Riemannian metric space. They generate a *unit mesh* (as defined hereafter) with respect to the Riemannian metric space.

First, we define the notion of unit element.

Definition 1. *A triangle (resp. tetrahedron) K is a unit element with respect to a metric $\mathcal{M}$ if the length $l_{\mathcal{M}}$ of all its edges $(\boldsymbol{e}_i)_{i\in[1,3]}$ (resp. $(\boldsymbol{e}_i)_{i\in[1,6]}$) is unit in $\mathcal{M}$:*

$$\forall i \in [1, n_{\boldsymbol{e}}],\ l_{\mathcal{M}}(\boldsymbol{e}_i) = 1 \tag{3.14}$$

with $n_{\boldsymbol{e}} = 3$ (resp. $n_{\boldsymbol{e}} = 6$).

The volume in the metric $|K|_{\mathcal{M}}$ and the Euclidean volume $|K|$ of a unit triangle in $\mathcal{M}$ are

$$|K|_{\mathcal{M}} = \frac{\sqrt{3}}{4} \tag{3.15}$$

$$|K| = \frac{\sqrt{3}}{4}\det(\mathcal{M}^{-1/2})\ , \tag{3.16}$$

and the volumes of a unit tetrahedron are

$$|K|_{\mathcal{M}} = \frac{\sqrt{2}}{12} \tag{3.17}$$

$$|K| = \frac{\sqrt{2}}{12} \det(\mathcal{M}^{-1/2}) \ . \tag{3.18}$$

The notion of *unit mesh* comes from the unit element. Strictly, a *unit mesh* is a mesh composed of unit elements with respect to a prescribed metric field. However, the existence of a mesh composed only with unit element is not guaranteed. For instance, a space as simple as $\mathbb{R}^3$ cannot be filled with unit tetrahedra with respect to the identity metric field. Thus, the constraint of unity is relaxed: a *quasi unit element* with respect to a metric $\mathcal{M}$ is such that the length of its edges verifies

$$\forall i \in [1, n_e], \ l_{\mathcal{M}}(\boldsymbol{e}_i) \in [\frac{1}{\sqrt{2}}, \sqrt{2}] \ . \tag{3.19}$$

In this context, a unit mesh is then defined as a mesh composed of quasi unit elements. To avoid the generation of null volume elements, a quality function is imposed in the mesh generators [122].

2 The continuous mesh framework

The continuous mesh framework is a theory which provides a continuous representation of a mesh. Its key point consists in the demonstration of a duality between a discrete mesh and its continuous representation. Once this duality is proved, the continuous interpolation error can be defined and minimised by solving an optimisation problem using mathematical tools in this continuous framework.

2.1 Duality discrete-continuous mesh

The correspondence between discrete and continuous elements has been developed for triangular/tetrahedral elements in [46, 122]. It lies on the previously defined notion of unit elements.

Local duality

In [122], a local metric is proved to be a class of equivalence of unit elements, as stated in the following proposition and illustrated in figure 3.2:

Proposition 1. *Let $\mathcal{M}$ be a metric tensor, there exists an infinite set of tetrahedra that are unit with respect to $\mathcal{M}$. Conversely, given an element $K = (\boldsymbol{e}_i)_{i\in[1,n_e]}$ such that $|K|_{I_3} \neq 0$, then there is a unique $\mathcal{M}$ for which element K is unit with respect to $\mathcal{M}$.*

The relation unit with respect to $\mathcal{M}$ defines a class of equivalence among the set of all discrete elements.

A continuous element and a discrete element are also connected as proved in [122]. In particular, the metric in which a discrete element is unit can be found from the geometrical

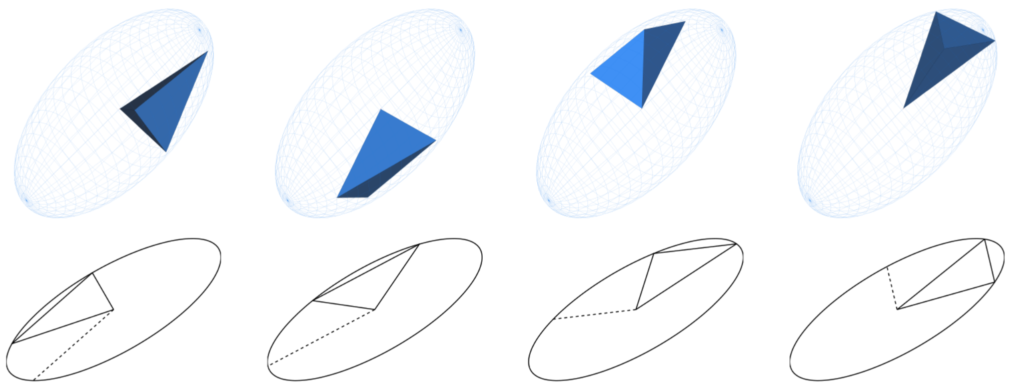


Figure 3.2: Examples of unit tetrahedra and triangles for a given metric represented by its unit ball (figure extracted from [122]).

properties of the discrete element in the Euclidean space. Using the orientations defined in figure 3.3, a 2D metric [46] writes

$$\mathcal{M} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \tag{3.20}$$

with

$$a = \frac{1}{||\boldsymbol{e}_1||_2^2} \tag{3.21}$$

$$b = \frac{||\boldsymbol{e}_3||_2^2 - ||\boldsymbol{e}_2||_2^2}{2\,||\boldsymbol{e}_1||_2^2\,||\boldsymbol{e}_1 \wedge \boldsymbol{e}_2||_2} \tag{3.22}$$

$$c = \frac{3\,||\boldsymbol{e}_1||_2^4 + (||\boldsymbol{e}_2||_2^2 - ||\boldsymbol{e}_3||_2^2)^2}{4\,||\boldsymbol{e}_1||_2^2\,||\boldsymbol{e}_1 \wedge \boldsymbol{e}_2||_2^2}\ . \tag{3.23}$$

and $(\boldsymbol{e}_i)_{i\in[\![1,3]\!]}$ are the edges of the triangular element, such as defined in figure 3.3.

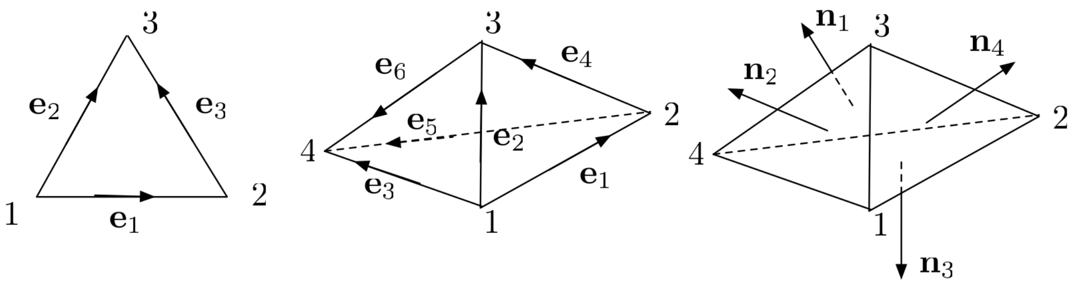


Figure 3.3: Conventions used to enumerate the edges and the faces of a triangle and of a tetrahedron. (figure extracted from [122])

Global duality

The equivalence class from Proposition 1 is only local therefore it must be extended to a whole mesh. This is achieved if we work in the Riemannian metric space $\boldsymbol{M}$. The main difficulty is to take into account the variations of the application $\boldsymbol{x} \mapsto \mathcal{M}(\boldsymbol{x})$. To achieve this, $\boldsymbol{M}$ is rewritten using its decomposition to distinguish its local and global properties [46, 122]:

$$\forall \mathbf{x} \in \Omega, \quad \mathcal{M}(\mathbf{x}) = d^{\frac{2}{3}}(\mathbf{x})\mathcal{R}(\mathbf{x}) \begin{pmatrix} r_1^{-\frac{2}{3}}(\mathbf{x}) & & \\ & r_2^{-\frac{2}{3}}(\mathbf{x}) & \\ & & r_3^{-\frac{2}{3}}(\mathbf{x}) \end{pmatrix} \mathcal{R}^T(\mathbf{x}) \ , \tag{3.24}$$

where

- h_i are the characteristic lengths deduced from the eigenvalues λ_i of the metric: $h_i = \sqrt{\frac{1}{\lambda_i}}$;
- d is the density of the metric: $d = \left(\prod_{i=1}^n h_i\right)^{-1} = \left(\prod_{i=1}^n \lambda_i\right)^{\frac{1}{2}}$;
- r_i are the anisotropic ratio coefficients: $r_i = h_i^3 \left(\prod_{i=1}^n h_i\right)^{-1}$;
- $\mathcal{R}$ contains the eigenvectors $(\boldsymbol{v}_i)$ of $\mathcal{M}$.

In this form, the density d is only responsible for the local accuracy of the metric, whereas the anisotropic coefficient r_i are responsible for the anisotropic properties of the metric. It is useful to notice that, by construction,

$$d = \sqrt{det(\mathcal{M})} \ . \tag{3.25}$$

The global level of accuracy of $\boldsymbol{M}$ is quantified by its *complexity* defined as:

$$\mathcal{C}(\mathbf{M}) = \int_\Omega d(\mathbf{x})\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\mathrm{d}\mathbf{x} \ . \tag{3.26}$$

This quantity can be interpreted as the continuous equivalent of the discrete number of elements or vertices of a discrete mesh.

Relation (3.24) shows the duality between meshes and Riemannian spaces. This duality is justified by the strict equivalence between the discrete quantities (orientation, stretching and size of an element, number of elements) and their continuous counterparts (respectively $\mathcal{R}$, r_i, d, $\mathcal{C}(\boldsymbol{M})$). Finally, a continuous mesh of a domain Ω is defined [46, 122] as a Riemannian metric space or equivalently as a set of continuous metric tensors $\boldsymbol{M} = (\mathcal{M}(\boldsymbol{x}))_{\boldsymbol{x}\in\Omega}$.

2.2 Continuous linear interpolate and continuous interpolation error

Once the mesh model $\boldsymbol{M}$ is defined, it can be used to derive a robust error estimate as shown in the work of A. Loseille and F. Alauzet [46, 122], where the use of the interpolation error is proposed as an error indicator for mesh refinement. Here, we briefly describe the expression of the interpolation error in this continuous framework and the equivalent discrete counterpart. We consider a continuous mesh $(\mathcal{M}(\boldsymbol{x}))_{\boldsymbol{x}\in\Omega}$ in a domain Ω, and u a quadratic positive function with H its associated Hessian matrix.

Continuous interpolation error

Let $\pi_{\mathcal{M}}$ be a linear interpolation operator in the continuous Riemannian metric space. We denote $\pi_{\mathcal{M}}u$ the continuous linear interpolate of the function u. Let $|u(\boldsymbol{x}) - \pi_{\mathcal{M}}u(\boldsymbol{x})|$ be the local continuous interpolation error associated to $\boldsymbol{x} \in \Omega$, and $||u - \pi_{\mathcal{M}}u||_{\boldsymbol{L}^1(\Omega)}$ the global continuous interpolation error.

By geometrical considerations – finding the plane which approximate at best u on the unity ball of $\mathcal{M}$, the local interpolation error is proved [46, 122] to write under the form

$$\forall \boldsymbol{x} \in \Omega,\ |u - \pi_{\mathcal{M}}u|(\boldsymbol{x}) = c_n \operatorname{tr}\left(\mathcal{M}(\boldsymbol{x})^{-\frac{1}{2}} H(\boldsymbol{x}) \mathcal{M}(\boldsymbol{x})^{-\frac{1}{2}}\right)\ , \tag{3.27}$$

with tr(·) the trace of a matrix, and $c_n \in \mathbb{R}$ a constant depending on the dimension n of the problem.

Discrete interpolation error

Consider now the discrete mesh $\mathcal{H}$ of the discrete domain Ω_h which is unit with respect to $(\mathcal{M}(\boldsymbol{x}))_{\boldsymbol{x}\in\Omega}$. Similarly to the continuous space, we introduce the linear interpolation operator Π_h and $\Pi_h u$ the discrete linear interpolate of u. Let K be a discrete element of Ω_h and $||u - \Pi_h u||_{\boldsymbol{L}^1(K)}$ be the discrete interpolation error on K.

By direct integration on an element of reference, the error on K is proved [46, 122] to be independent of the shape of the simplicial element and writes

$$||u - \Pi_h u||_{\boldsymbol{L}^1(K)} = \frac{\sqrt{3}}{64} \det(\mathcal{M}^{-\frac{1}{2}}) \operatorname{tr}\left(\mathcal{M}^{-\frac{1}{2}} H \mathcal{M}^{-\frac{1}{2}}\right) \tag{3.28}$$

in 2D, and

$$||u - \Pi_h u||_{\boldsymbol{L}^1(K)} = \frac{\sqrt{2}}{240} \det(\mathcal{M}^{-\frac{1}{2}}) \operatorname{tr}\left(\mathcal{M}^{-\frac{1}{2}} H \mathcal{M}^{-\frac{1}{2}}\right) \tag{3.29}$$

in 3D.

Relation between discrete and continuous interpolation error

Introducing the expression of the volume of an element $|K|$ (in equations (3.6), (3.15) and (3.17)), the discrete and the continuous error are linked through the following proposition [46, 122].

Proposition 2. *In the neighborhood of a point $\boldsymbol{a}$ in Ω, let u_Q be the quadratic approximation of the function u. Then, for every unit element K with respect to $\mathcal{M}(\boldsymbol{a})$, there exists a unique function $\pi_{\mathcal{M}}u$ which verifies :*

$$\forall \mathbf{a} \in \Omega, \quad |u - \pi_{\mathcal{M}}u|\,(\mathbf{a}) = \frac{\|u_Q - \Pi_h u_Q\|_{L^1(K)}}{|K|} = c_n \operatorname{tr}\left(\mathcal{M}(\mathbf{a})^{-\frac{1}{2}} H(\mathbf{a}) \mathcal{M}(\mathbf{a})^{-\frac{1}{2}}\right)\ , \tag{3.30}$$

with $c_n = c_3 = \frac{1}{20}$ in 3D and $c_n = c_2 = \frac{1}{16}$ in 2D.

This proposition is extended to non-quadratic twice continuously differentiable functions by considering their absolute Hessian $|H|$ obtained by taking the absolute eigenvalues of their Hessian H.

The discrete and continuous global interpolation errors are then defined as :

$$||u - \Pi_h u||_{\boldsymbol{L}^1(\Omega_h)} = \sum_{K \in \mathcal{H}} ||u - \Pi_h u||_{\boldsymbol{L}^1(K)} \tag{3.31}$$

$$||u - \pi_{\mathcal{M}} u||_{\boldsymbol{L}^1(\Omega)} = \int_{\Omega} |u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} \tag{3.32}$$

In theory, the equality between equations (3.31) and (3.32) holds only when the mesh is unit with respect to a constant metric tensor and u is quadratic. In practice, it has been observed in [46] and [123] that (3.31) $\approx$ (3.32) even for non-quadratic functions and non-constant continuous meshes.

3 Continuous interpolation error for quad/octree grids

The previous section provides the expression of the continuous interpolation error for a mesh composed with unit triangles/tetrahedra with respect to a Riemannian metric field $\boldsymbol{M}$. However, as described in chapter 2, a mesh obtained with Basilisk is only composed of square or cubic elements. In this section, we will apply this theory to quad/octree grids, for which the extension is not automatic.

3.1 Quadtree continuous interpolation error

We extend the definition of unit element to the polyhedron as in [80]:

Definition 2. *A polyhedron $K = (\boldsymbol{e}_i)_{i \in n_{\boldsymbol{e}}}$ with n_e edges is unit with respect to a metric $\mathcal{M}$ if the length of all of its edges is unit in this metric.*

In 2D, a square element with a length h is unit for the metric $\mathcal{M}_u = h^{-2} I_2$. We want to verify if the error estimator developed for simplicial elements in equation (3.30) is also correct for a square element considering its metric $\mathcal{M}_u$. To do so, we propose in this section a demonstration based on the geometrical decomposition of the square element into triangular elements, which has the advantage to be easily illustrated. Another proof which directly computes the interpolation error on a square of reference is presented in appendix A and will be used for the 3D case in the next section.

We consider and subdivide a square element K into n_T triangular subdomains $(K_i)_{i \in [1, n_T]}$ (a partition of K), as illustrated in figure 3.4:

$$K = \bigcup_{i \in [1, n_T]} K_i \tag{3.33}$$

$$\forall \ (i,j) \in [1, n_T]^2, \ i \neq j \ \Rightarrow \ K_i \cap K_j = \varnothing \ . \tag{3.34}$$

The proof consists as follow. First, we determine the metric in which the triangles are unit. Then, we estimate the interpolation error $||u - \Pi_h u||_{K_i}$ on these triangles, and finally, we use it to find the error on the square which is the sum of the error on each triangles

$$||u - \Pi_h u||_K = \sum_{i \in [1, n_T]} ||u - \Pi_h u||_{K_i} \ . \tag{3.35}$$

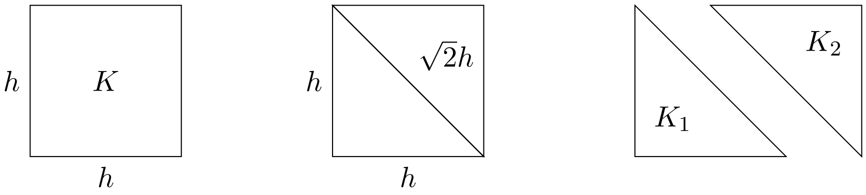


Figure 3.4: Partition of the square element K into two triangles K_1 and K_2.

Let consider the triangular element K_1 in figure 3.4. Its metric (*cf.* equation (3.20)) is defined by

$$\mathcal{M}_{K_1} = h^{-2}\begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix} \tag{3.36}$$

and admits the decomposition $\mathcal{M}_{K_1} = \mathcal{R}_{K_1}\Lambda_{K_1}\mathcal{R}_{K_1}^T$ with the eigenvalues matrix

$$\Lambda_{K_1} = h^{-2}\begin{pmatrix} \frac{3}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \tag{3.37}$$

and the normalized eigenvectors

$$\mathcal{R}_{K_1} = \frac{\sqrt{2}}{2}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} . \tag{3.38}$$

We introduce the transformation

$$\mathcal{M}_{K_1}^{-1/2} = \mathcal{R}_{K_1}\Lambda_{K_1}^{-1/2}\mathcal{R}_{K_1}^T = \frac{h}{2\sqrt{3}}\begin{pmatrix} \sqrt{2}+\sqrt{6} & \sqrt{2}-\sqrt{6} \\ \sqrt{2}-\sqrt{6} & \sqrt{2}+\sqrt{6} \end{pmatrix} , \tag{3.39}$$

and we pose the Hessian matrix H of a positive quadratic function u

$$H = \begin{pmatrix} a & b \\ b & c \end{pmatrix} . \tag{3.40}$$

Using equation (3.28) and the Euclidian volume of a triangular element (equation (3.15)), we obtain:

$$\begin{aligned} ||u - \Pi_h u||_{\boldsymbol{L}^1(K_1)} &= \frac{4}{3}\, c_2\, |K_1|\, h^2(a + c - b) \\ &= \frac{4}{3}\, c_2\, \frac{h^2}{2}\, h^2(a + c - b) \end{aligned} \tag{3.41}$$

with $c_2 = \frac{1}{16}$.

We perform the same analysis with the element K_2. Its metric is given by

$$\mathcal{M}_{K_2} = h^{-2}\begin{pmatrix} 1 & -1/2 \\ -1/2 & 1 \end{pmatrix} , \tag{3.42}$$

and from that, we obtain the error on K_2:

$$\begin{aligned} ||u - \Pi_h u||_{\boldsymbol{L}^1(K_2)} &= \frac{4}{3}\, c_2\, |K_2|\, h^2 (a + c + b) \\ &= \frac{4}{3}\, c_2\, \frac{h^2}{2}\, h^2 (a + c + b) \ . \end{aligned} \tag{3.43}$$

Finally, we estimate the error on the square element K using the relation (3.35):

$$\begin{aligned} ||u - \Pi_h u||_{\boldsymbol{L}^1(K)} &= ||u - \Pi_h u||_{\boldsymbol{L}^1(K_1)} + ||u - \Pi_h u||_{\boldsymbol{L}^1(K_2)} \\ &= \frac{4}{3}\, c_2\, h^2\, h^2\, (a + c) \\ &= \frac{4}{3}\, c_2\, |K|\, h^2\, (a + c) \\ &= c_2\, |K|\, \mathrm{tr}\left(\mathcal{M}_K^{-1/2} H \mathcal{M}_K^{-1/2}\right) \end{aligned} \tag{3.44}$$

with $c_2 = \frac{1}{16}$ and

$$\mathcal{M}_K = \frac{3}{4}\, h^{-2}\, I_2 \ . \tag{3.45}$$

Written under this form, the error on K (equation (3.44)) has the same form than the error on a triangular element (equation (3.27)), but the metric to consider is not the metric in which the square is unit. It leads to the definition of the *equivalent metric*: the metric representative for the polyhedron in terms of error. This metric is not the metric in which the element is unit, and account for the fact that the element is not a simplicial element, but a polyhedron.

The error on the square element can also be written depending on the metric in which it is unit $\mathcal{M}_u = h^{-2} I_2$ as

$$||u - \Pi_h u||_{\boldsymbol{L}^1(K)} = C_2\, |K|\, \mathrm{tr}\left(\mathcal{M}_u H \mathcal{M}_u\right) \tag{3.46}$$

with $C_2 = \frac{1}{12}$. Under this form, the difference between the error on triangles and square elements is contained in the prefactor C_2, which is different from the prefactor $c_2 = \frac{1}{16}$ valid for triangular elements.

Finally, similarly as in equation (3.30) (following [123]), we define the continuous interpolate $\pi_{\mathcal{M}} u$ such that the continuous error $|u - \pi_{\mathcal{M}} u|$ is equal to the error on a square element K represented by its equivalent metric $\mathcal{M}_K$:

$$\begin{aligned} \forall\, \boldsymbol{x} \in \mathbb{R}^2,\ |u - \pi_{\mathcal{M}} u| &= \frac{||u - \Pi_h u||_{\boldsymbol{L}^1(K)}}{|K|} \qquad (3.47) \\ &= c_n\, \mathrm{tr}\left(\mathcal{M}_K^{-1/2} H \mathcal{M}_K^{-1/2}\right) \ . \qquad (3.48) \end{aligned}$$

As for the case of the triangle, this result is extended to non-quadratic functions by considering their absolute Hessian $|H|$ obtained from their Hessian H by taking its absolute eigenvalues.

3.2 Octree continuous interpolation error

In this section, we derive a $\boldsymbol{L}^1$-norm interpolation error estimate for a cubic element K represented by its vertex $(\boldsymbol{v}_i)_{i\in[\![1,8]\!]}$ as it was done ine the work of Loseille and Alauzet [123] for a tetrahedral element. To do so, we show off an exact point-wise error estimate of the interpolation error within an element of reference K_{ref}. We apply a change of variables to obtain the error on K, from the error on K_{ref}.

The reference element is a cubic element with unit edge lengths. Its vertices are noted $\hat{\boldsymbol{v}}_1 = (0,0,0)$, $\hat{\boldsymbol{v}}_2 = (1,0,0)$, $\hat{\boldsymbol{v}}_3 = (0,1,0)$, $\hat{\boldsymbol{v}}_4 = (0,0,1)$, $\hat{\boldsymbol{v}}_5 = (1,1,0)$, $\hat{\boldsymbol{v}}_6 = (1,0,1)$, $\hat{\boldsymbol{v}}_7 = (0,1,1)$, $\hat{\boldsymbol{v}}_8 = (1,1,1)$. The transformation between a coordinate $\boldsymbol{x} \in K$ and $\hat{\boldsymbol{x}} \in K_{ref}$ is given by

$$\boldsymbol{x} = \boldsymbol{v}_1 + h\hat{\boldsymbol{x}} \tag{3.49}$$

with h the edge length of K.

Let $u(\boldsymbol{x}) = \boldsymbol{x}^T H \boldsymbol{x}$ be a quadratic function exactly valued at the vertices of the cube and represented by its associated Hessian $H = \begin{pmatrix} a & d & e \\ d & b & f \\ e & f & c \end{pmatrix}$. We pose Π_h the trilinear interpolate operator. In the framework of K_{ref}, u reads

$$u(\boldsymbol{x}(\hat{\boldsymbol{x}})) = \frac{1}{2}\boldsymbol{v}_1^T H \boldsymbol{v}_1 + \frac{1}{2}\boldsymbol{v}_1^T H h\hat{\boldsymbol{x}} + \frac{1}{2}\hat{\boldsymbol{x}}^T h H \boldsymbol{v}_1 + \frac{1}{2}h^2 \hat{\boldsymbol{x}}^T H \hat{\boldsymbol{x}} \ . \tag{3.50}$$

Linear and constant terms of $u(\boldsymbol{x}(\hat{bx}))$ are exactly interpolated, thus we consider, without loss in generality, only the quadratic term $\hat{u}(\boldsymbol{x}) = \frac{1}{2}h^2\hat{\boldsymbol{x}}^T H \hat{\boldsymbol{x}}$, as $(u - \Pi_h u)(\boldsymbol{x}) = (\hat{u} - \Pi_h \hat{u})(\boldsymbol{x})$. To improve readability, we write u instead of $\hat{u}$ in the following.

For all $\boldsymbol{x} = (x,y,z)^T \in K_{ref}$, the trilinear interpolate $\Pi_h u$ of the function u is

$$\Pi_h u(\boldsymbol{x}) = \alpha + \lambda x + \mu y + \eta z + \beta xy + \gamma xz + \omega yz \ . \tag{3.51}$$

As the solution is known at the vertices, we have

$$\begin{cases} \Pi_h u(\boldsymbol{v}_1) & = \alpha = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_1)) = 0 \\ \Pi_h u(\boldsymbol{v}_2) & = \lambda = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_2)) = \frac{1}{2}ah^2 \\ \Pi_h u(\boldsymbol{v}_3) & = \mu = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_3)) = \frac{1}{2}bh^2 \\ \Pi_h u(\boldsymbol{v}_4) & = \eta = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_4)) = \frac{1}{2}ch^2 \\ \Pi_h u(\boldsymbol{v}_5) & = \lambda + \mu + \beta = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_5)) = \frac{1}{2}(a+b+2d)h^2 \\ \Pi_h u(\boldsymbol{v}_6) & = \lambda + \eta + \gamma = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_6)) = \frac{1}{2}(a+c+2e)h^2 \\ \Pi_h u(\boldsymbol{v}_7) & = \mu + \eta + \omega = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_7)) = \frac{1}{2}(b+c+2f)h^2 \\ \Pi_h u(\boldsymbol{v}_8) & = \lambda + \mu + \eta + \beta + \gamma + \omega = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_8)) = \frac{1}{2}(a+b+c+2d+2e+2f)h^2 \end{cases}$$

From that, we readily obtain the point-wise interpolate

$$\Pi_h u(\boldsymbol{x}) = \frac{1}{2}ah^2 x + \frac{1}{2}bh^2 y + \frac{1}{2}ch^2 z + dh^2 xy + eh^2 xz + fh^2 yz \ . \tag{3.52}$$

and the point-wise interpolation error on the reference element K_{ref}

$$(u - \Pi_h u)(\boldsymbol{x}) = \frac{1}{2} h^2 \left[a(x^2 - x) + b(y^2 - y) + c(z^2 - z) \right] \ . \tag{3.53}$$

By direct integration, we obtain the interpolation error on K_{ref} in $\boldsymbol{L}^1$-norm

$$\begin{aligned} ||u - \Pi_h u||_{\boldsymbol{L}^1(K_{ref})} &= \int_0^1 \int_0^1 \int_0^1 |u - \Pi_h u|(\boldsymbol{x}) dx\, dy\, dz && (3.54) \\ &= \frac{1}{12} \operatorname{tr}(H) h^2 && (3.55) \end{aligned}$$

We apply the change of variables between a coordinate $\boldsymbol{x} \in K$ and $\hat{\boldsymbol{x}} \in K_{ref}$. It writes

$$\int_K f(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} = \int_{K_{ref}} f(\hat{\boldsymbol{x}}) h^3 \mathrm{d}\hat{\boldsymbol{x}} = |K| \int_{K_{ref}} f(\hat{\boldsymbol{x}}) \mathrm{d}\hat{\boldsymbol{x}} \tag{3.56}$$

with $|K| = h^3$ the volume of the element K. Thus, the $\boldsymbol{L}^1$-norm interpolation error of the element K writes

$$||u - \Pi_h u||_{\boldsymbol{L}^1(K)} = \frac{1}{12} \operatorname{tr}(H) h^2 |K| \ . \tag{3.57}$$

Similarly as in two dimension, we define the continuous interpolate $\pi_{\mathcal{M}} u$ such that the continuous error $|u - \pi_{\mathcal{M}} u|$ is equal to the error on a square element K:

$$\begin{aligned} \forall \boldsymbol{x} \in \mathbb{R}^2, \ |u - \pi_{\mathcal{M}} u| &= \frac{||u - \Pi_h u||_{\boldsymbol{L}^1(K)}}{|K|} && (3.58) \\ &= c_3 \operatorname{tr} \left(\mathcal{M}_K^{-1/2} H \mathcal{M}_K^{-1/2} \right) && (3.59) \\ &= C_3 \operatorname{tr} \left(\mathcal{M}_u H \mathcal{M}_u \right) \ , && (3.60) \end{aligned}$$

with $c_3 = \frac{1}{20}$, $C_3 = \frac{1}{12}$, $\mathcal{M}_u = h^{-2} I_3$ the metric in which the cube is unit, and $\mathcal{M}_K = \frac{3}{5}\, h^{-2}\, I_3$ the equivalent metric with I_3 the identity matrix.

3.3 Uniformly refined mesh interpolation error

From the continuous error in equation (3.48), the $\boldsymbol{L}^p$ interpolation error on a uniformly refined mesh composed with squared elements is readily obtained. It writes

$$\begin{aligned} ||u - \pi_{\mathcal{M}} u||_{\boldsymbol{L}^p(\Omega)} &= \left(\int_\Omega \left(|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) \right)^p \ \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{p}} \\ &= c_n \left(\int_\Omega \operatorname{tr} \left(\mathcal{M}_K^{-1/2}(\boldsymbol{x}) |H|(\boldsymbol{x}) \mathcal{M}_K^{-1/2}(\boldsymbol{x}) \right)^p \ \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{p}} \\ &= C_n\, h^2 \left(\int_\Omega \operatorname{tr} \left(|H|(\boldsymbol{x}) \right)^p \ \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{p}} \ , && (3.61) \end{aligned}$$

with h the uniform cell size, and $C_2 = C_3 = \frac{1}{12}$.

Introducing the number of element N of the uniform mesh given by

$$N = \left(\frac{L_0}{h}\right)^n \ , \tag{3.62}$$

with L_0 the size of a square domain and n the dimension, the interpolation error on a uniform mesh reduces to

$$||u - \pi_{\mathcal{M}} u||_{\boldsymbol{L}^p(\Omega)} = C_n \, L_0^2 \left(\int_\Omega \operatorname{tr}(|H|(\boldsymbol{x}))^p \, \mathrm{d}\boldsymbol{x}\right)^{\frac{1}{p}} N^{-\frac{2}{n}} \ . \tag{3.63}$$

Validation

We validate the quadtree uniform error estimation using the quadratic function

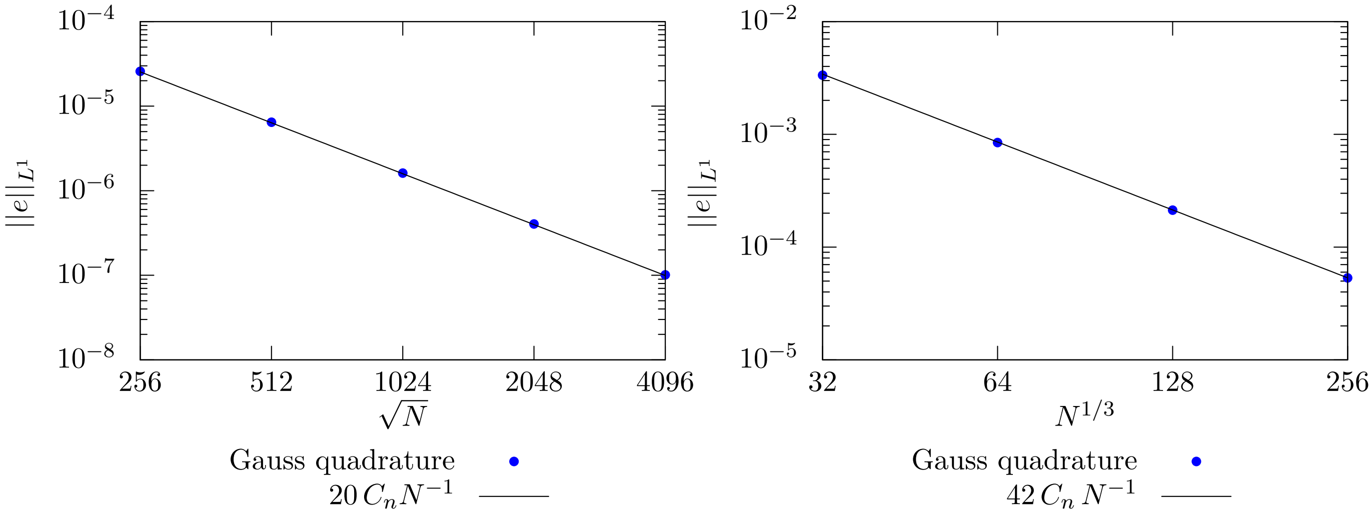
$$u(x, y) = 6x^2 + 2xy + 4y^2 \tag{3.64}$$

with the associated Hessian $H = \begin{pmatrix} 12 & 2 \\ 2 & 8 \end{pmatrix}$.

Imposing a unit length square domain, the global error reduces to

$$||u - \pi_{\mathcal{M}} u||_{\boldsymbol{L}^1(\Omega)} = 20 \, C_2 \, N^{-1} \ . \tag{3.65}$$

In figure 3.5a, we can see that the theoretical prediction perfectly matches the error measured with a Gauss quadrature. This behaviour is confirmed for various functions in table 3.1, where we introduce $c_{th} = C_n \int_\Omega \operatorname{tr}(|H|(\boldsymbol{x})) \, \mathrm{d}\boldsymbol{x}$ and c_{exp} the constant obtained from the linear fit of the error measured with the Gauss quadrature. This validates the theoretical development made for the quadtree mesh.



(a) Error convergence for the function u in equation (3.64)

(b) Error convergence for the function u_{3D} in equation (3.66)

Figure 3.5: Comparison between the error measured on uniform meshes for a function and its theoretical prediction.

Similarly, we present in figure 3.5b the validation of the octree estimation using the quadratic function

$$u_{3D}(x, y) = 6x^2 + 2xy + 4y^2 + 3xz + 4yz + 11z^2 \tag{3.66}$$

with the associated Hessian $H_{3D} = \begin{pmatrix} 12 & 2 & 3 \\ 2 & 8 & 4 \\ 3 & 4 & 22 \end{pmatrix}$. Its global interpolation error reduces to

$$||u_{3D} - \pi_{\mathcal{M}} u_{3D}||_{\boldsymbol{L}^1(\Omega)} = 42\, C_3\, N^{-1} \ . \tag{3.67}$$

$u(x, y)$	c_{th}	c_{exp}
$6x^2 + 2xy + 4y^2$	1.67	1.67
e^{2x^2+y}	4.57	4.57
$\sin(x)$	0.038	0.038
$x^2 - e^{2(x-1)/0.1}$	1.73	1.73

Table 3.1: Comparison between the theoretical error prefactor and the measured error prefactor for different functions.

4 Optimal metric-based mesh adaptation

4.1 L^p-norm anisotropic metric-based interpolation error

In section 2, we described the continuous mesh framework. In this framework the continuous interpolation error is defined and shown to be equivalent to the discrete interpolation error on a discrete mesh. However, if these approach are equivalent, the continuous mesh offers the opportunity to use more mathematical tools, in particular the calculus of variation, the differentiation and the optimization.

In the current section, we use these tools to derive the *a priori* metric-based anisotropic interpolation error as described in [46, 117, 30, 118]. It provides the anisotropic metric field which minimizes the interpolation error in $\boldsymbol{L}^p$ norm. In this context, find an adapted mesh is seen as a constrained minimization problem : it consists to find a mesh which minimizes the error of a given sensor on a given domain. We present here only the key steps of the demonstration of this estimate. The complete demonstration is available in the work of Loseille and Alauzet [46, 123].

Error model

To define the minimization problem, an error model is first proposed. Let $\boldsymbol{M} = (\mathcal{M}(\boldsymbol{x}))_{\boldsymbol{x}\in\Omega}$ be a continuous mesh in $\mathbb{R}^n$, with its local eigenvectors $(\boldsymbol{v}_i)_{i\in[\![1,n]\!]}$. Let u be a twice differentiable function whose Hessian is H and positive Hessian $|H|$ – as a reminder, $|H|$ is obtained from H by taking its absolute eigenvalues.

From equation (3.30), the following error model is proposed

$$|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) = c_n \sum_{i=1}^{n} h_i^2(\boldsymbol{x}) \left| \boldsymbol{v}_i^T(\boldsymbol{x}) H(\boldsymbol{x}) \boldsymbol{v}_i(\boldsymbol{x}) \right| \, , \tag{3.68}$$

with $c_n = c_3 = \frac{1}{20}$ in 3D and $c_n = c_2 = \frac{1}{16}$ in 2D, and $(h_i)_{i \in [\![1,n]\!]}$ are the cell size in each direction.

Minimization problem

The global error is minimized in $\mathbf{L^p}$ norm. This choice provides a multiscale error estimate, as we will see later. The minimisation problem writes :

$$\text{Find} \min_{\boldsymbol{M}} \left(\int_\Omega \left(|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) \right)^p \, \mathrm{d}\boldsymbol{x} \right) \, , \tag{3.69}$$

$$\text{under the constraint } \mathcal{C}(\boldsymbol{M}) = \int_\Omega \left(\prod_{i=1}^{n} h_i(\boldsymbol{x}) \right)^{-1} \mathrm{d}\boldsymbol{x} = N_* \, , \tag{3.70}$$

with N_* the fixed complexity constrain. The constraint is necessary to avoid the trivial solution $(h_i)_{i \in [\![1,n]\!]} = (0)_{i \in [\![1,n]\!]}$, which leads to a null error.

Resolution key points

This optimization problem is solved through several steps whose key ideas are the following. First, the minimisation problem is rewritten using the metric decomposition (3.24) based on the density and the anisotropic quotients. One major advantage of these variables is that the constraint becomes linear leading to a convex optimization problem. Then, the problem is solved through the theory of Lagrange multipliers. At this step, the optimal metric size is obtained. It is then proved to be unique. The last step is the obtention of the orientation of the metric, by observing that the error is minimal when its eigen vectors are aligned with whose of the Hessian $|H|$.

Solution of the minimization problem

Once the problem is solved, the optimal metric is given by [46, 123]

$$\mathcal{M}_{L^p}(\mathbf{x}) = N_*^{\frac{2}{n}} \left(\int_\Omega \det\left(|H(\mathbf{x})|\right)^{\frac{2p}{2p+n}} \, \mathrm{d}\boldsymbol{x} \right)^{-\frac{2}{n}} \det\left(|H(\mathbf{x})|\right)^{\frac{-1}{2p+n}} |H(\mathbf{x})| \tag{3.71}$$

where:

- N_* is the complexity. In the asymptotic regime, N_* is linearly dependant with the number of vertex [123] ;
- n is the dimension of the problem ;
- p is the norm in which the error is minimized ($\boldsymbol{L^p}$ norm);
- $|H|$ is the absolute Hessian of the sensor.

The local and global optimal error respectively write

$$|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) = c_n \, n \, N_*^{-\frac{2}{n}} \left(\int_\Omega (det(|H(\boldsymbol{x})|)^{\frac{p}{2p+n}} \ \mathrm{d}\boldsymbol{x} \right)^{\frac{2}{n}} (det(|H(\boldsymbol{x})|))^{\frac{1}{2p+n}} \tag{3.72}$$

$$\|u - \pi_{\mathcal{M}} u\|_{\mathbf{L^p}} = \left(\int_\Omega |u - \pi_{\mathcal{M}} u|^p(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{p}} = c_n \, n \, N_*^{-\frac{2}{n}} \left(\int_\Omega (det(|H(\boldsymbol{x})|)^{\frac{p}{2p+n}} \ \mathrm{d}\boldsymbol{x} \right)^{\frac{2p+n}{np}} . \tag{3.73}$$

The local metric is proportional to the factor $(det(|H(\boldsymbol{x})|))^{\frac{-1}{2p+n}}$, which evolves with p : this term is more sensible to small fluctuations in the solution when p tends to 1. On the opposite, when p tends to infinity, it is less sensitive to the small perturbations. This property justify the “multiscale” appellation originally used for the AMR method based on this result. It is today more commonly classified as “feature-based” as it based on a feature (the solution Hessian) of a flow field.

This global error is proportional to the factor $N^{-\frac{2}{n}}$, which leads to a global second order convergence: $N^{-\frac{1}{n}}$ is indeed proportional to the mean cell size $\bar{h}$ (see equation (3.62)), and so, $N^{-\frac{2}{n}}$ is proportional to $\bar{h}^2$.

4.2 $\mathbf{L^p}$-norm isotropic interpolation error for quad/octree grids

In the previous section, the anisotropic metric based error estimate [46, 123] is recalled. But as seen in chapter 1, the Basilisk grid structure is by nature completely isotropic and composed with square (or cubic) elements instead of simplicial elements. It is then interesting to particularize the anisotropic error estimate into an isotropic error estimate for the quad/octree grid structure. This estimate is obtained following the same demonstration steps than its anisotropic counterpart presented in [30, 46, 122, 123].

Error model

Let consider a set of metric $(\mathcal{M}(\boldsymbol{x}))_{\boldsymbol{x}\in\Omega}$ in $\mathbb{R}^n$. We note $(\boldsymbol{v}_i)_{i\in[\![1,n]\!]}$ the eigenvectors of $\mathcal{M}$. Let u be a twice differentiable function whose Hessian is H and positive Hessian $|H|$.

A similar error model to the anisotropic case in equation (3.68) is proposed:

$$|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) = C_n \sum_{i=1}^{n} h^2(\boldsymbol{x}) \, \boldsymbol{v}_i^T(\boldsymbol{x}) |H|(\boldsymbol{x}) \boldsymbol{v}_i(\boldsymbol{x}) \ , \tag{3.74}$$

with $C_2 = C_3 = \frac{1}{12}$ and h is the metric size in each direction. It takes into account the isotropic nature of the studied quad/octree grids, and its decomposition into square/cubic elements, as described in section 3 of this chapter.

In order to simplify the notations, we introduce $(\boldsymbol{\gamma}_i)_{i\in[\![1,n]\!]}$ such that the error writes

$$|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) = C_n \, h^2(\boldsymbol{x}) \sum_{i=1}^{n} \boldsymbol{\gamma}_i(\boldsymbol{x}) \ . \tag{3.75}$$

Minimization problem

The global error is minimized in $\mathbf{L^p}$ norm. This choice provides a feature-based error estimate. The minimization problem writes :

$$\text{Find} \min_{\boldsymbol{M}} \left(\int_\Omega \left(|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) \right)^p \ \mathrm{d}\boldsymbol{x} \right) , \tag{3.76}$$

$$\text{under the constraint } \mathcal{C}(\boldsymbol{M}) = \int_\Omega \left(h^n(\boldsymbol{x}) \right)^{-1} \ \mathrm{d}\boldsymbol{x} = N_*. \tag{3.77}$$

The resolution of the optimization problem is based on a change of variable. As the searched metric is isotropic, the anisotropic quotients r_i defined in the anisotropic case are no more necessary. Only the local density of the metric, define as $d = h^{-n}$, is used. After introducing this change of variable, we readily obtain the following minimization problem:

$$\text{Find} \min_{\boldsymbol{M}} \left(\int_\Omega \left(d^{-\frac{2}{n}}(\boldsymbol{x}) \sum_{i=1}^{n} \boldsymbol{\gamma}_i(\boldsymbol{x}) \right)^p \ \mathrm{d}\boldsymbol{x} \right) , \tag{3.78}$$

$$\text{under the constraint } \mathcal{C}(\boldsymbol{M}) = \int_\Omega d(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} = N_* \ . \tag{3.79}$$

Solution

The problem is solved using Lagrange multipliers theory. The detailed demonstration is described in appendix B. It results that the continuous error writes

$$|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) = C_n \, N^{-\frac{2}{n}} \left(\int_\Omega \left(\operatorname{tr}(|H|(\boldsymbol{x})) \right)^{\frac{np}{2p+n}} \ \mathrm{d}\boldsymbol{x} \right)^{\frac{2}{n}} \left(\operatorname{tr}(|H|(\boldsymbol{x})) \right)^{\frac{n}{2p+n}} \ , \tag{3.80}$$

with $C_2 = C_3 = \frac{1}{12}$, and

$$||u - \pi_{\mathcal{M}} u||_{\Omega, L^p} = C_n \, N^{-\frac{2}{n}} \left(\int_\Omega \left(\operatorname{tr}(|H|(\boldsymbol{x})) \right)^{\frac{np}{2p+n}} \ \mathrm{d}\boldsymbol{x} \right)^{\frac{2p+n}{np}} \ . \tag{3.81}$$

where:

- N is the number of elements ;
- n is the dimension of the problem ;
- p is the norm in which the error is minimized ($\boldsymbol{L}^p$ norm) ;
- $|H|$ is the absolute Hessian of the sensor ;
- $\operatorname{tr}(A)$ denotes the trace of the matrix A.

One of the major interest of this estimator is that it allows L^p norm mesh optimisation. This helps to deal with multiscale solutions : In L^1 norm, all the scales of the solution should be captured whereas in L^p norms, when p tends to infinity, the focus should be made on the scales with the highest amplitude.

It is particularly noticeable that the total error equation (3.81) takes the form

$$||u - \pi_{\mathcal{M}} u||_{\Omega, L^p} = C_{opt} \, N^{-\frac{2}{n}} \ , \tag{3.82}$$

with C_{opt} a fully determined constant only depending on the Hessian of the sensor and the error norm. Thus, this result is well suited to estimate the optimal AMR performances one could expect for a particular sensor. It will be used as a reference in the rest of this thesis.

4.3 Adaptive Algorithm

The Basilisk quad/octree grid structure and the grid refinement/coarsening operations already exist, as seen in chapter 2. In order to perform multiscale mesh adaptation, we combine them with the quad/octree error estimate developed in section 4.2. At the end of the adaptive procedure, we want to obtain a mesh reaching a prescribed objective cell number N_{obj} which equidistributes the error in all cells.

For an analytical solution, a loop is necessary to obtain an adapted mesh as the refinement/coarsening procedure of Basilisk allows only one level of refinement/coarsening per step. The mesh is then converged following the algorithm illustrated on algorithm 3.

Algorithm 3: Adaptation loop for analytical functions.

```
Set the initial mesh $\mathcal{M}_0$ (coarse uniform Cartesian mesh);
Count the number of elements $N$;
Set the initial threshold $\varepsilon$;
Set the maximum authorized level $l_{max}$;
Set the objective number of elements $N_{obj}$;
while $N \notin N_{obj} \pm 5\%$ do
    Compute the solution $S_i$ on $\mathcal{M}_i$ ;
    Compute the error $||e||_{l,\boldsymbol{L^p}}$ in each element;
    Restrict it to estimate the error $||e||_{l-1,\boldsymbol{L^p}}$ in the parent cell;
    foreach element do
        if $||e||_{l,\boldsymbol{L^p}} > \varepsilon$ and $l \leq l_{max}$ then
            Refine the cell;
        else
            if $||e||_{l-1,\boldsymbol{L^p}} < \varepsilon/4$ and neighboring cells are not too fine then
                Coarsen the cell;
            end
        end
    end
    Count the number of elements $N$ and update $\varepsilon$;
end
```

The initial solution S_0 is obtained on an initial mesh $\mathcal{M}_0$ – usually a coarse uniform mesh. The isotropic $\mathbf{L^p}$-norm error $||u - \pi_{\mathcal{M}} u||_{K,\boldsymbol{L^p}}$ is then computed in each mesh element K and compared with an ε criterion. If the error is greater than ε, the cell is refined. Else, if the error in the parent cell (detailed hereafter) is lower than ε, the cell is coarsen. Else the cell remains unchanged. Finally, the current number of elements is compared with the objective cell number N_{obj}, and ε is updated. The loop continue until N_{obj} is reached.

The error $||e||_{l-1,\mathbf{L^p}}$ in a parent cell with level $l-1$ is obtained through a restriction operation from the error $||e||_{l,\mathbf{L^p}}$ in its children which, by construction, have their level equal to l. In particular, for $\mathbf{L^p}$-norm error, we have

$$||e||_{l-1,\mathbf{L^p}} = \left(\sum_{children} ||e||_{l,\mathbf{L^p}}^p \right)^{\frac{1}{p}} \tag{3.83}$$

Note that the user-imposed constraint on the maximum level authorized to avoid over-refinement (*cf.* chapter 2) still exists. For the results presented in the section 5, it is put to a value high enough to avoid any influence.

5 Quad/Octree optimal error validation

In this section, we present and discuss some results obtained with isotropic interpolation error based adaptive method. The main objective is to validate the implementation of this error estimate and the corresponding AMR procedure. To do so, we compare the L^p norm interpolation error of two analytical solutions on different meshes: L^p-norm isotropic adapted meshes, wavelet-based adapted meshes and uniformly refined meshes. The global error on the analytical solution is computed using Gauss quadrature methods.

5.1 Multiscale example

We first validated the isotropic interpolation error estimation on the multiscale sinusoidal function $u_1(x, y)$, illustrated in appendix C:

$$u_1(x, y) = \sin(a\,(xy - b))\left(0.01 + e^{-(d\,(xy-c))^2}\right) \tag{3.84}$$

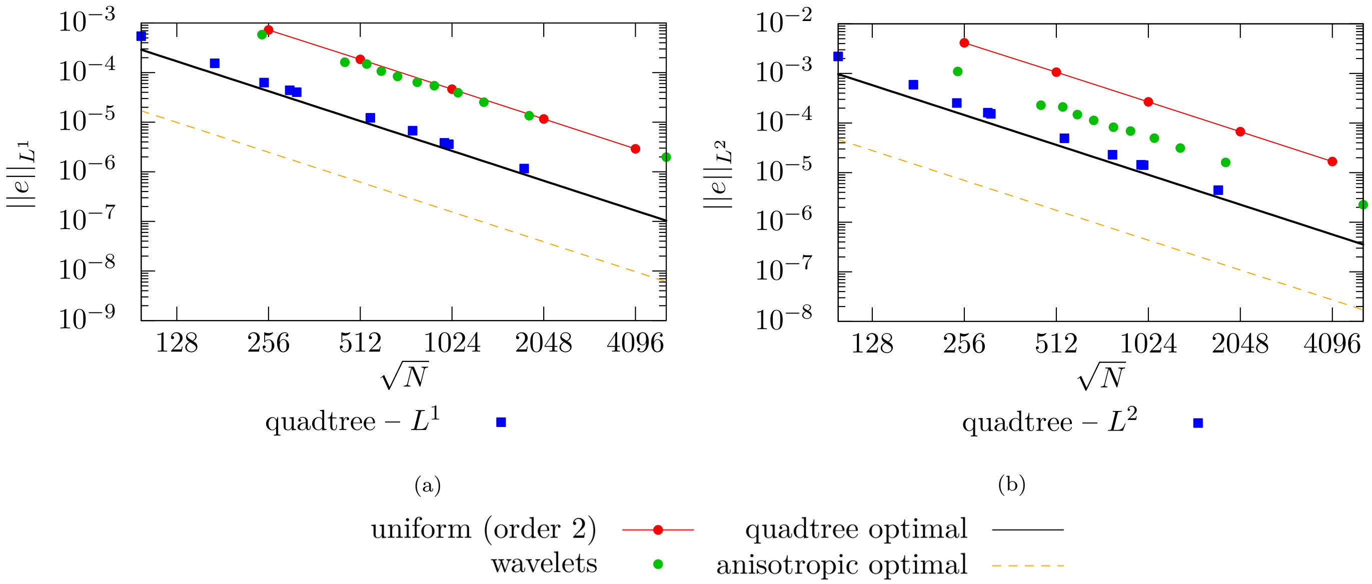
with $a = 5\pi$, $b = 1.5\pi/50$, $c = 3\pi/50$ and $d = 50\pi$.

This choice – inspired from but not identical to [46, 30] – is motivated by its multiscale aspect, with two scales in amplitude and two scales in wavelength. This function has been exact valued at cell centers (no equation is solved in this section). The AMR results presented hereafter are obtained using the isotropic interpolation error estimator on this field. They are compared with uniformly refined mesh results, and the wavelet-based AMR method.

Two convergence results are presented in figure 3.6: the first one (figure 3.6a) shows the global error convergence curve in $\boldsymbol{L}^1$-norm , whereas the second one (figure 3.6b) presents the global error convergence curve in $\boldsymbol{L}^2$-norm. The isotropic-based adapted meshes are done considering the corresponding $\boldsymbol{L}^p$-norm. These convergence curves compares the global error – in $\boldsymbol{L}^1$ or $\boldsymbol{L}^2$ norm – *vs* the square root of the cell number, which is equivalent to the inverse of an average cell size. These graphs show interesting results.

First, consider the figure 3.6a. Five elements are presented on this figure. The first two are the metric-based optimal error estimates developed in the previous section : the isotropic one equation (3.81) is represented with a continuous black line, and the anisotropic one equation (3.73) is represented with a continuous orange line. It can be seen that for this particular function, in both norms $\boldsymbol{L}^1$ and $\boldsymbol{L}^2$, an anisotropic mesh could have much better performance than an isotropic mesh, with around 10 times less error for the same cell number. This observation is consistent with previous observations in literature [124, 125].

The three other convergence curve on this figure are obtained with uniformly refined meshes (red dots and line), wavelet-based AMR method (green dots) and $\boldsymbol{L}^1$ isotropic adaptive procedure (blue squares). The convergence with uniformly refined meshes is presented to check the efficiency of the AMR methods : bigger gaps between uniformly refined curve and AMR curves implies higher efficiencies of the adaptive method. In this case, we observe



(a)

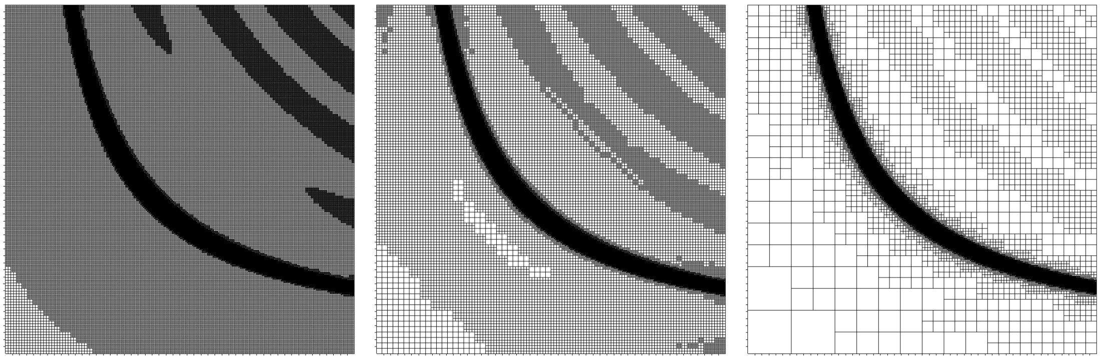
(b)

Figure 3.6: L^p-norm global error of the multiscale sinusoidal function u_1 *vs* square root of cell number (inversely proportional to the mean cell size). These error convergence curves compare the results obtained with $\boldsymbol{L}^p$ isotropic AMR method and theory, wavelet-based Basilisk AMR method, and uniformly refined meshes. Left: $\boldsymbol{L}^1$ norm AMR. Right: $\boldsymbol{L}^2$ norm AMR.

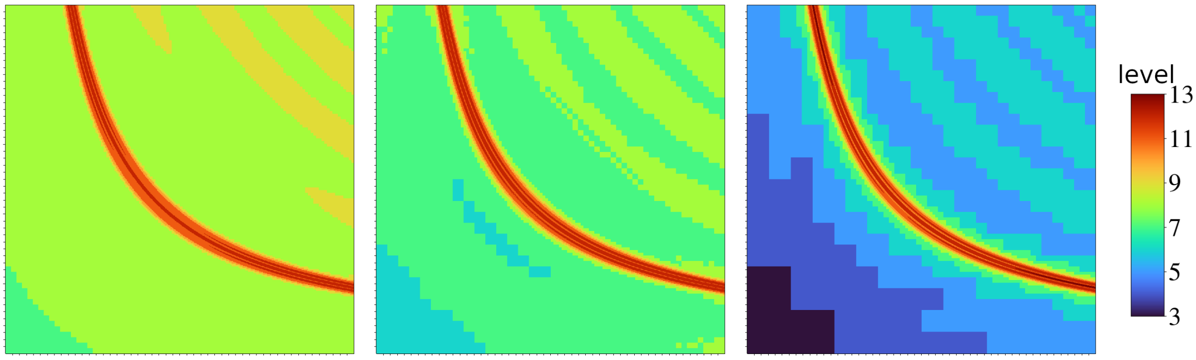
that for a given number of elements, the wavelet-based AMR method of Basilisk is as efficient as uniformly refined meshes, whereas $\boldsymbol{L}^1$ isotropic AMR method is much more efficient, as it reduces the error by around one decade. This shows that the $\boldsymbol{L}^1$ isotropic adaptive procedure is an improvement compare to the wavelet-based AMR method. Finally, concerning the error measured on $\boldsymbol{L}^1$ isotropic AMR meshes, we clearly see that it follows the theoretical isotropic optimal error. This validates our AMR implementation and it shows that it is not necessary to take into account all of constraints relative to Basilisk meshes (see chapter 2, section 1), as taking into account the isotropic constraint and the square form of the elements is enough to obtain good numerical results.

The analysis of the figure 3.6b is almost identical to the analysis of the figure 3.6a, except that $\boldsymbol{L}^2$ error and $\boldsymbol{L}^2$ isotropic meshes are considered instead of their $\boldsymbol{L}^1$ counterparts. The results are very similar: the main difference is that the error on wavelet-based adapted meshes is lower than the error on uniformly refined meshes. Thus, the wavelet-based AMR method is more efficient when the error is checked in $\boldsymbol{L}^2$ norm than it is in $\boldsymbol{L}^1$ norm. Otherwise, the $\boldsymbol{L}^2$ isotropic AMR errors compares also well with their theoretical prediction, which validates it in $\boldsymbol{L}^2$ norm too.

The meshes obtained with the three AMR method (L^p-norm isotropic based, and wavelet-based) are shown figure 3.7. They contained approximately 600 000 cells. The $\boldsymbol{L}^1$-norm isotropic metric based AMR mesh is the most uniform. That means that it has the highest element size ratio between its finest and coarsest cells $\boldsymbol{C}_{r,\boldsymbol{L}^1} = \frac{h_{fine}}{h_{coarse}} = \frac{2^7}{2^{12}} \approx 3.1 \times 10^{-2}$. As a comparison, this ratio is $\boldsymbol{C}_{r,\boldsymbol{L}^2} = \frac{2^6}{2^{12}} \approx 1.6 \times 10^{-2}$ for the $\boldsymbol{L}^2$-norm based adapted mesh, and $\boldsymbol{C}_{r,Basilisk} = \frac{2^3}{2^{13}} \approx 9.8 \times 10^{-4}$ for the wavelet-based mesh. This is logical, as on a theoretical point of view, it should be able to capture all the scales of the solution equally. The L^2-norm



(a) $\boldsymbol{L}^1$-norm AMR (b) $\boldsymbol{L}^2$-norm AMR (c) Wavelet-based AMR



(d) $\boldsymbol{L}^1$-norm AMR (e) $\boldsymbol{L}^2$-norm AMR (f) Wavelet-based AMR

Figure 3.7: Mesh and level field obtained for three AMR for the multiscale sinusoidal function u_1: $\boldsymbol{L}^1$-norm isotropic metric-based meshes, $\boldsymbol{L}^2$-norm isotropic metric-based meshes, and wavelet-based Basilisk adapted meshes. Each mesh contains around 600 000 cells.

adapted mesh gives more importance to high amplitude scales than the L^1-norm adapted mesh. These observations are similar to the observations done with anisotropic metric based AMR method in [46, 30]. Finally, the Basilisk adaptive method provides a mesh which is finer than the two others in the region of highest amplitude, and coarser in the rest of the domain, which is expected due to the mathematical properties of the *wavelet* methods [84]. This implies that the obtained meshes are consistent with the global error convergence curves figure 3.6.

	error L^1	error L^2	error L^∞
mesh L^1	$5.8\ 10^{-6}$	$2.5\ 10^{-5}$	$6.2\ 10^{-4}$
mesh L^2	$7.2\ 10^{-6}$	$2.2\ 10^{-5}$	$5.9\ 10^{-4}$
mesh wavelets	$5.6\ 10^{-5}$	$8.1\ 10^{-5}$	$9.9\ 10^{-4}$

Table 3.2: Error in L^1, L^2 and L^∞ norm on the three meshes figure 3.7 for u_1.

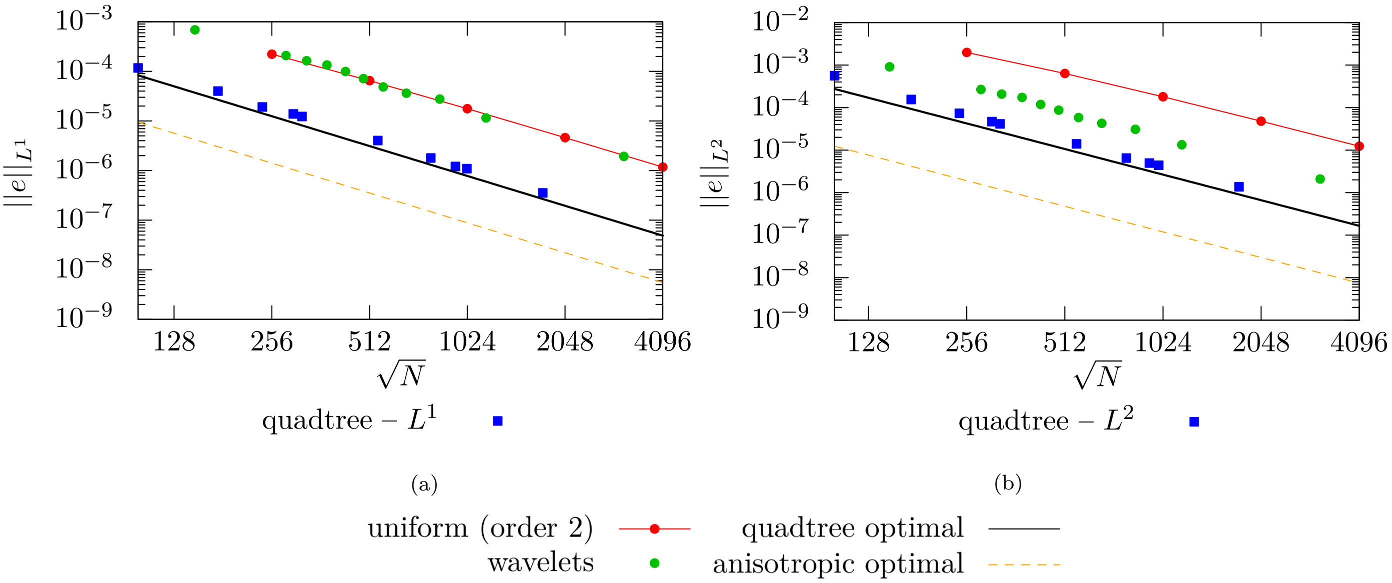
The table 3.2 shows the error obtained in three norms ($\boldsymbol{L}^1$, $\boldsymbol{L}^2$ and $\boldsymbol{L}^\infty$) on the three meshes presented on figure 3.7. It confirms the results obtained with isotropic metric based AMR: the $\boldsymbol{L}^1$ error is minimal on the $\boldsymbol{L}^1$ AMR mesh and it equidistributes the most the contribution of all solution scales. This translates in terms of cell size distribution : the $\boldsymbol{L}^1$ mesh is the most uniform between the compared meshes. On the same way, the $\boldsymbol{L}^2$ error is minimized on the $\boldsymbol{L}^2$ mesh. Finally, for this particular example, the wavelet-based AMR leads to the highest error in the three considered norms.

5.2 Boundary layer example

We validated the isotropic interpolation error estimation on a second test case proposed by John in [126] and used by Hauke [127] to assess the performance of adaptive methods. This case depends on a coefficient κ and posseses typical regular boundary layers at $x = 1$ and $y = 1$ when κ tends to 0. It is mathematically described by the function

$$u_2(x, y) = xy^2 - y^2 \exp(\frac{2(x-1)}{\kappa}) - x \exp(\frac{3(y-1)}{\kappa}) + \exp(\frac{2(x-1)+3(y-1)}{\kappa}) \ , \qquad (3.85)$$

with $\kappa = 10^{-2}$.

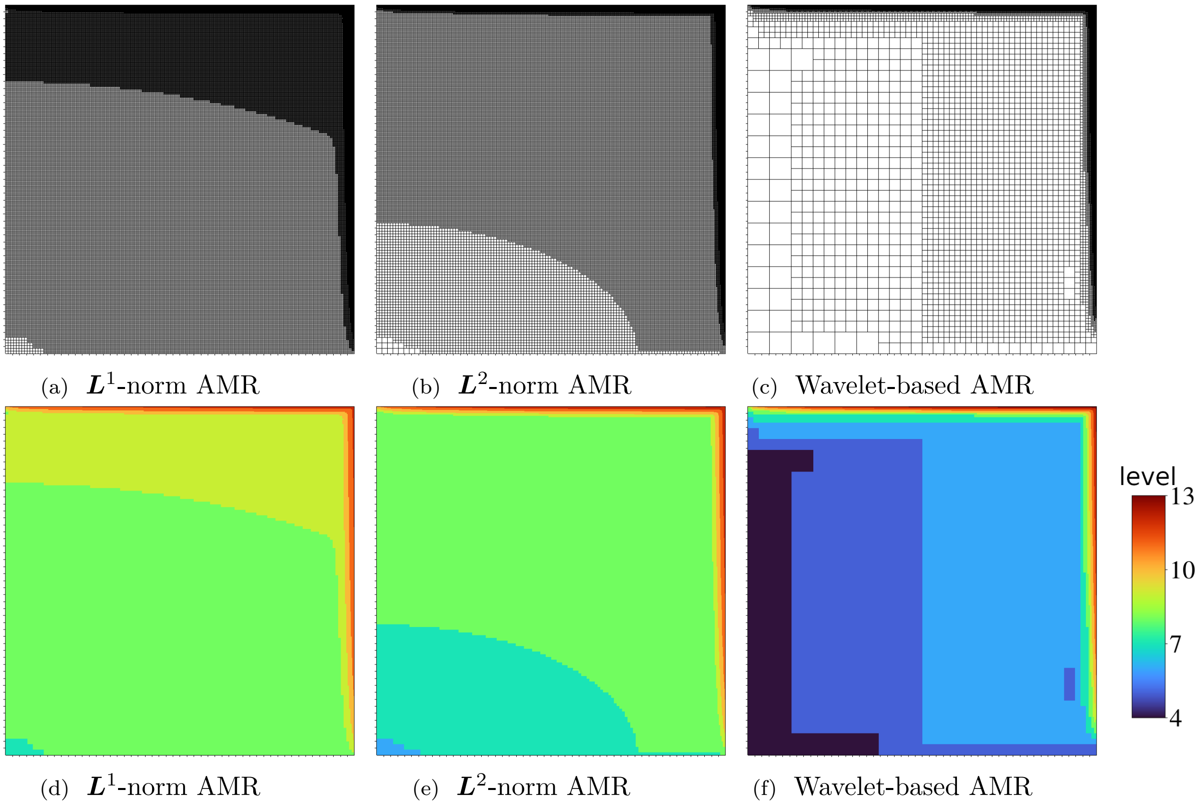


(a)

(b)

Figure 3.8: L^p-norm global error of the boundary layer function u_2 *vs* square root of cell number (inversely proportional to the mean cell size). These error convergence curves compare the results obtained with $\boldsymbol{L^p}$ isotropic AMR method and theory, wavelet-based Basilisk AMR method, and uniformly refined meshes. Left: $\boldsymbol{L^1}$ norm AMR. Right: $\boldsymbol{L^2}$ norm AMR.

As in the previous paragraph, we compare the results obtained with the $\boldsymbol{L^p}$-norm isotropic adaptive method and the wavelet-based AMR procedure. The error convergence curves are shown on figure 3.8, and the 300 000 elements meshes are presented on figure 3.9. Even if this solution has only one scale, we can draw exactly the same analyses and conclusions than for the multiscale case in paragraph 5.1. In particular, the error obtained on meshes obtained with the $\boldsymbol{L^p}$ norm-driven AMR are follows the theoretical optimal expectations, and the obtained cell size range is directly dependent on the method used. These observations validate the $\boldsymbol{L^p}$ isotropic AMR method and its sensitivity to the choice of the $\boldsymbol{L^p}$ norm.



(a) $\boldsymbol{L}^1$-norm AMR (b) $\boldsymbol{L}^2$-norm AMR (c) Wavelet-based AMR

(d) $\boldsymbol{L}^1$-norm AMR (e) $\boldsymbol{L}^2$-norm AMR (f) Wavelet-based AMR

Figure 3.9: Mesh and level field obtained for three AMR for the boundary layer function u_2: $\boldsymbol{L}^1$-norm isotropic metric-based meshes, $\boldsymbol{L}^2$-norm isotropic metric-based meshes, and wavelet-based Basilisk adapted meshes. Each mesh contains around 600 000 cells.

6 Conclusion

In this chapter, we introduced the continuous mesh framework. This framework, based on Riemannian metric spaces, allows the definition of a continuous mesh and a continuous interpolation error, which leads to the use of mathematical continuous tools available in optimization theory. These continuous quantities are equivalent to their discrete counterparts, and are the basis of anisotropic metric based mesh adaptation, whose main concept is to find the optimal mesh which minimizes the continuous interpolation error of the solution in $\boldsymbol{L}^p$-norm.

Once this theory is recalled, we have extended it for the quad/octree grid structure of the software Basilisk. In particular, this grid structure imposes isotropic meshes composed with square/cubic elements, which leads to a modified interpolation error estimation in comparison with the case of anisotropic meshes composed of simplicial elements. Then, the $\boldsymbol{L}^p$-norm optimal interpolation error for quad/octree grids is derived and is used as a local indicator for mesh refinement. The refinement/coarsening method of Basilisk is adapted and combined with the new error estimate. This metric-based approach also provides explicitly – without undetermined constants – the total interpolation error of an analytical function on the optimal adapted mesh.

Finally, the new AMR method is validated on analytical cases. It is proved to be sensitive to the $\boldsymbol{L}^p$ norm chosen for the adaptation and to find the optimal mesh in the considered norm. For the test-cases, this method is an upgrade to the existing classical *wavelet-based* mesh adaptation used in Basilisk.

4| A novel AMR method for elliptic equations

In the previous chapter, we developed an interpolation error estimate and used it as a refinement criterion to drive mesh adaptation. This method provides accurate and efficient adapted meshes for analytical functions. However, in daily computational fluid dynamics, one has to deal with numerical solutions which contain an accumulation of different errors.

This chapter provides an AMR algorithm to reduce numerical errors when solving the Poisson-Helmholtz equation. Using as reference the expression for the optimal AMR convergence curve previously obtained, the behaviour of the numerical error is discussed for different grids. The role of the ratio between the minimum grid size and the averaged grid size is shown to be critical to describe the behavior of the numerical error observed. Only for sufficiently smooth transition of the grid size, the total numerical error can be modelled as a local interpolation error. Based on these results, a new AMR algorithm is proposed by imposing additional constrains in the method. The resulting AMR method is validated and discussed on various test-cases.

Contents

1 Numerical error quantification for elliptic equations 54

1.1 Experiment and first observations: a coarsening study 54

1.2 Definition of AMR performance quantities 56

2 A simple error model for PDE equations 63

3 Mesh compression ratio constrain 64

3.1 Estimation methods for the compression ratio constrain 65

3.2 Security coefficient 72

4 Proposed AMR strategy 74

4.1 AMR algorithm 74

4.2 Illustrative example 74

5 Validation 76

5.1 Boundary layer 77

5.2 Multiscale sinus 79

5.3 Potential flow around a circular cylinder 82

6 Conclusion 84

1 Numerical error quantification for elliptic equations

1.1 Experiment and first observations: a coarsening study

The behavior of the numerical errors obtained on a numerical solution is not trivial. On a general case, the numerical errors depend on the nature of the equation being solved, the discretization method and the characteristics of the grid being used. In this section we use an example to discuss the behavior of the numerical error in the Basilisk solver. In particular, we show to which extend the total error obtained from a numerical solver can be correlated to the local interpolation error. This is of course a desirable feature, as any known local representation of the numerical error allows for an easy implementation of local and fast AMR algorithms.

We start considering our elliptic equation of reference: the Poisson-Helmholtz

$$\nabla \cdot (D\nabla u) + \lambda u = s \ , \tag{4.1}$$

where the source term s is computed such that the solution u is equal to a multiscale sinusoidal function $u_1(x,y) = \sin(a\,(xy-b))\left(0.01 + e^{-(d\,(xy-c))^2}\right)$ with $a = 5\pi$, $b = 1.5\pi/50$, $c = 3\pi/50$ and $d = 50\pi$. (see appendix C). In the current study, we impose $D = 1$ and $\lambda = -1$.

We perform a coarsening study illustrated in figure 4.1. First, an initial uniform mesh of level l_{max} is set and the solution is calculated. Then, a series of adapted mesh are obtained by coarsening the cells containing the smallest interpolation error. On each mesh, the numerical solution is computed by solving the discrete Poisson-Helmholtz equation alongside with the total numerical error simply obtained as the difference between the numerical and the analytical solution. At the same time, the analytical cell-centered solution is used to compute the interpolation error. Thus, we can monitor the evolution of both, numerical and interpolation errors, when reducing the number of elements. We first focus on the total error, and we use next the interpolation error as a comparison.

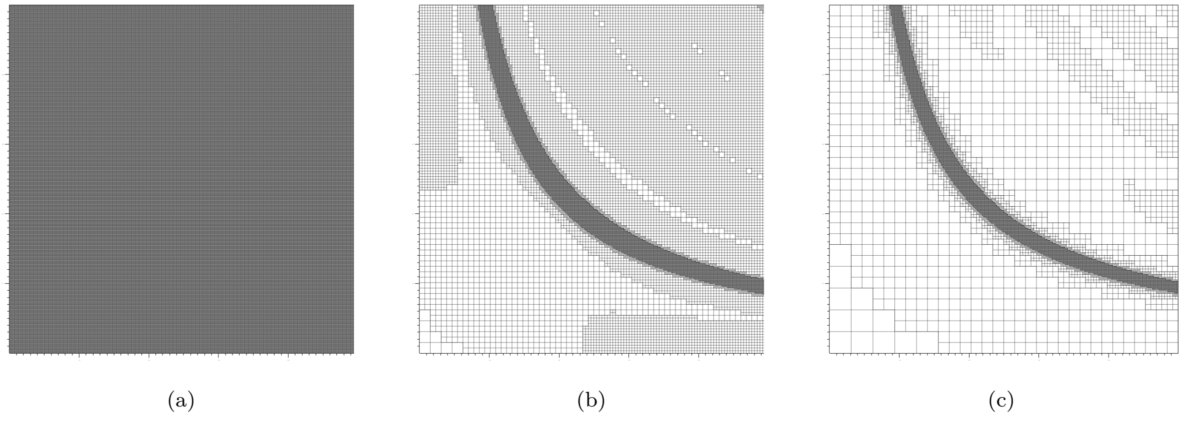
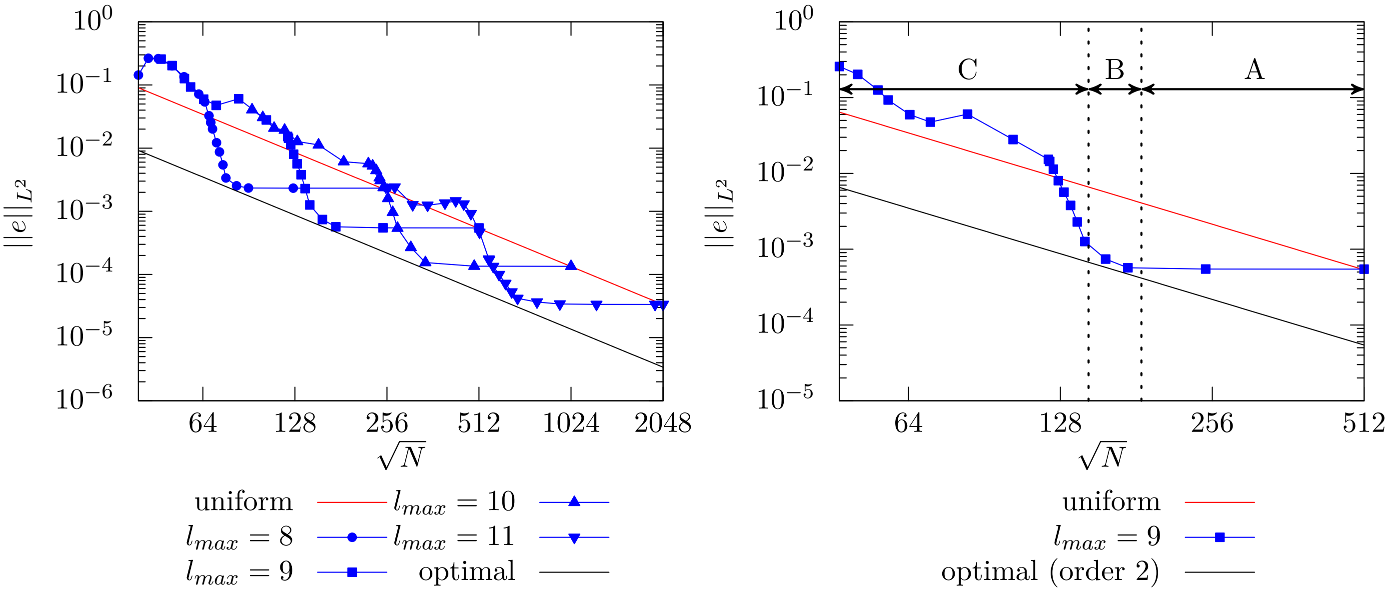

(a) (b) (c)

Figure 4.1: Illustration of the coarsening study: from the initial uniform mesh (a), the cells with the least interpolation error are coarsen. This operation is repeated and provides successively the mesh (b) and (c).

We perform this study for l_{max} between 8 and 11 and obtain the total numerical error convergence curves in figure 4.2a. On this graph, we compare the numerical error obtained with a uniform grid (plain red curve), the isotropic metric-based estimation of the optimal interpolation error convergence curve (plain black curve), and the evolution of the total error on the grids resulting from the coarsening study. Three regimes can be distinguished as illustrated in figure 4.2b. When the grid is almost uniform and most of the elements reach the imposed minimum grid size, the total error is almost insensitive to the number of elements. In this case most of the error is contained in the cells which have reached the minimum grid size criterion and one would need to reduce the minimum grid size in order to reduce further the error. This regime spans until the error is close to the minimal interpolation error curve. In this region, the interpolation error seems to be a good estimation of the total numerical error. After a small region where the numerical error is close to the optimal convergence curve, the total numerical error increases significantly faster than the optimal interpolation error indicating that the error is no longer well represented by the local interpolation error only. This last regime is case dependent, but it proves the existence of solutions in which the direct minimization of interpolation errors can be counter-productive in comparison with uniform grids. For instance, this naive AMR procedure leads to different grids when fixing the number of elements that depend on the maximum level of refinement allowed. Remarkably, the smaller the minimum grid size is allowed, the poorer is the performance of the grid obtained. This observation holds true, in this example, for all levels of refinement (*cf.* figure 4.2a). The possible existence of this last regime is problematic for AMR simulations and motivates the development of more careful AMR strategies to minimize the error.



(a) Coarsening study results for $l_{max} \in \llbracket 8, 11 \rrbracket$. For each l_{max}, the total error behaves similarly.

(b) For a decreasing number of elements: A: constant error. B: intermediate regime. C: dramatic error increase (faster than order 2).

Figure 4.2: Total error (blue) obtained for a coarsening study: an initially uniform mesh (red) is successively coarsen based on the interpolation error. This is repeated from several uniform meshes (Left). Right: two main error regimes A and C are observed, separated by a regime of transition B, which shows performances similar to the theoretical optimum (black). The error in the regime C becomes larger than the error on uniform meshes.

1.2 Definition of AMR performance quantities

The previous results of the coarsening study are now further analyzed. Let's introduce the **compression ratio** η_N as

$$\eta_N = \frac{N}{N_{h_{min}}} \ , \tag{4.2}$$

where N is the number of cells of an adapted mesh. Denoting h_{min} the minimal cell size of the adapted mesh, $N_{h_{min}}$ is the number of cells that would contain a uniform mesh whose cell size is h_{min}. The compression ratio represents the compression level of the AMR compared to uniform refinement (see figure 4.3) and by definition it is bounded in the interval $\eta_N \in]0,1]$. The limit $\eta_N = 1$ corresponds to a uniform mesh, whereas η_N tends to 0 as long as the number of element decreases while keeping a constant minimal cell size.

The compression ratio may be rewritten as a function of a cell size ratio. Noting $\overline{h}$ as the mean cell size of a N element adapted mesh, $\overline{h}$ and N are linked by the relation $N = \left(\frac{L_0}{\overline{h}}\right)^n$, with L_0 the domain size and n the dimension. Using the number of grid points for a uniform mesh with the minimum element size, $N_{h_{min}} = \left(\frac{L_0}{h_{min}}\right)^n$, we directly deduce that the compression ratio writes

$$\eta_N = \left(\frac{h_{min}}{\overline{h}}\right)^n \ . \tag{4.3}$$

We can also define the **loss** factor (*cf.* figure 4.3) L

$$L = \frac{||e||}{||e||_{opt}} \ , \tag{4.4}$$

where $||e||$ is the total error obtained on an adapted mesh containing N cells, and $||e||_{opt}$ is the minimal interpolation error theoretically predicted on a N cells mesh. This loss factor represents the lost of efficiency of the AMR method in terms of error for a given number of cells. In particular, assuming that the minimal interpolation error is representative of the minimal numerical error obtainable, the loss factor gives a measure of the increase of the amount of error obtained with the current grid with respect to the interpolation error obtained with the optimal grid with the same number of grid points. By definition, the loss factor verifies $L \geq 1$. The limit $L = 1$ is an optimum and it implies that the total error on the adapted mesh is equal to the theoretical minimal interpolation error for the same number of elements.

Finally, an alternative measure of the AMR performance is the **gain** factor (*cf.* figure 4.3)

$$G = \frac{||e||_{h_{u,N}}}{||e||} \ . \tag{4.5}$$

It represents the gap between the error $||e||_{h_{u,N}}$ on a uniform mesh containing N elements of size $h_{u,N}$ and the error on the current mesh $||e||$. G quantifies the gain in efficiency of an adapted mesh in comparison with a uniformly refined mesh for a given number of elements N.

The loss factor L and the gain factor G are inversely proportional: in chapter 3 we have seen that the error for a cartesian mesh can be expressed as $||e||_{h_{u,N}} = C_u \, N^{-\frac{2}{n}}$, while the

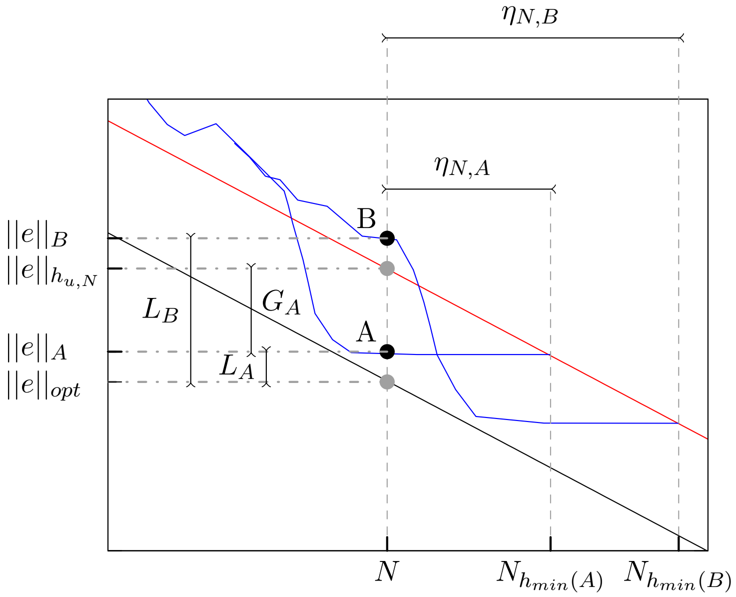


Figure 4.3: Compression ratio η_N, loss factor L and gain factor G visualisation for two different N elements meshes A and B having they own minimal cell size $h_{min}(A)$ and $h_{min}(B)$.

optimal error is $||e||_{opt} = C_o N^{-\frac{2}{n}}$, with C_u and C_o two constants depending on the Hessian of the solution. Thus, from the definition of L and G, we directly obtain

$$L = \frac{C_u}{C_o}\frac{1}{G} \ . \tag{4.6}$$

As the loss factor L and the gain factor G contains similar information, we focus on L for the current experiment. The gain factor G will be used later in this thesis.

We represent in figure 4.4 the loss factor as a function of the compression ratio from the coarsening study presented above (*cf.* figure 4.2a). The blue points correspond to the total error measured on the numerical solution, and the red points to the error on the cell-centered analytical solution. Several observations are interesting. First, the curves for all l_{max} superpose each other. That means that the loss factor and the compression ratio are independent from the level of refinement and therefore, they are well suited to describe the behavior of the error.

Secondly two regimes are clearly distinguishable for the total numerical error. For values of the compression ratio larger than a given critical value the loss factor is reduced while reducing the compression ratio. That means that the mesh is getting closer to the optimal one. Then, an optimum is clearly identifiable ($\eta_{N,opt} \approx 0.11$), where the loss factor is nearly equal to 1. A further decrease in the compression ratio leads to a fast increase in the loss factor, which corresponds to what we call regime of error explosion. This second regime is completely different for the error of the analytical solution, where the loss factor remains almost constant and equal to 1 for values of the compression ratio smaller than $\eta_{N,opt} \approx 0.11$. This is in agreement with the results of the chapter 3: the minimal interpolation error obtainable is well estimated through the metric based theory.

Finally, we clearly observe that the interpolation error is greater for the optimal compression ratio $\eta_{N,opt}$ than for lower compression ratios. Thus, the meshes which minimize the

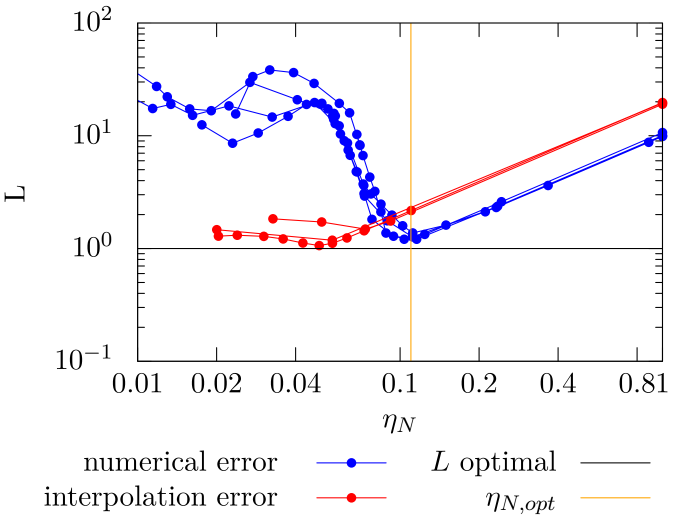
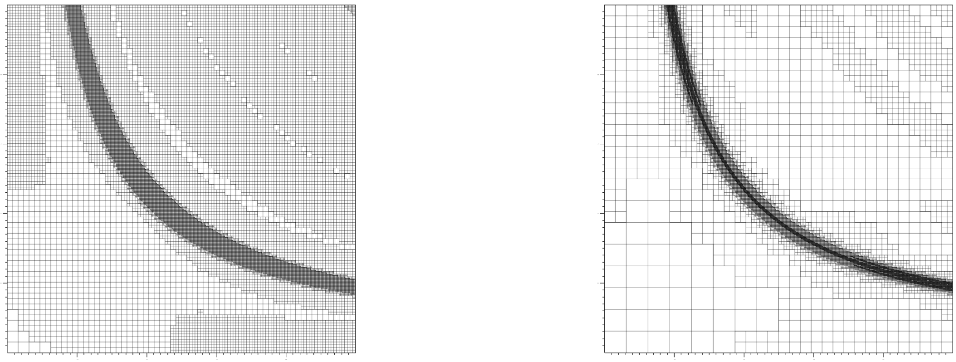


Figure 4.4: Loss factor *vs* compression ratio representation for the coarsening study. Blue: loss factor representative for the total error of the numerical solution. Red: loss factor representative for the interpolation error. The curves for the different coarsening studies ($l_{max} \in [\![8, 11]\!]$) superpose each other. For the total error, an almost optimal loss factor $L \approx 1.4$ is obtained for the corresponding optimal compression ratio $\eta_{N,opt} \approx 0.11$.

interpolation error are not necessarily the meshes which minimize the total error. In other words, reducing the interpolation error to reduce the total error (first error regime) is correct until a given point characterized by the optimal compression ratio. The corresponding meshes have a reduced interpolation error in comparison with uniformly refined meshes, but they don't minimize it – even if the reduced interpolation error is of the order of magnitude of the minimal interpolation error.

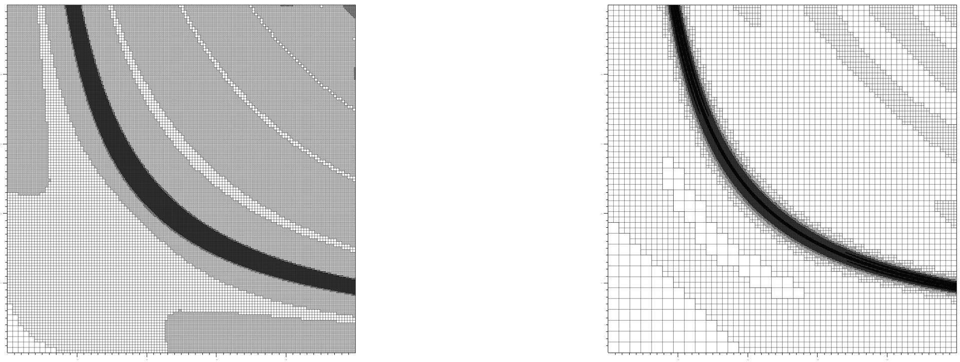
Figure 4.5 shows the meshes obtained for $\frac{\eta_N}{\eta_{N,opt}} \approx 1$, where the loss factor $L \approx 1.4$ is near the optimal, and $\frac{\eta_N}{\eta_{N,opt}} \approx 0.3$ (in the error explosion regime with $L > 10$) for different values of the minimal cell size. The corresponding normalized cell size distributions are shown in figure 4.6a, and the normalized cell area distributions are displayed in figure 4.6b. Comparing figure 4.5a and 4.5c, we visually see that the only difference between the two optimal meshes ($\frac{\eta_N}{\eta_{N,opt}} \approx 1$, $L \approx 1.4$) is that the mesh 4.5c is twice finer than the mesh 4.5a. This is quantitatively verified by the cell size and area distribution in figure 4.6a and 4.6b, where the two normalized distributions for $\frac{\eta_N}{\eta_{N,opt}} \approx 1$ (blue) are almost identical.

The two nearly optimal meshes ($\frac{\eta_N}{\eta_{N,opt}} \approx 1$, $L \approx 1.4$) are different from the two meshes lying in the error explosion regime ($\frac{\eta_N}{\eta_{N,opt}} \approx 0.3$, $L > 10$). In figure 4.5a we see that the two $L > 10$ meshes are composed with finer elements than the optimal meshes in the region of highest amplitude of the solution, and coarser elements in the rest of the domain. The normalized cell size and area distributions in figure 4.6a and 4.6b confirm this observation. In particular the normalized cell size distribution presents two different peaks localized at a level $l - l_{max} = -2$ and $l - l_{max} = 0$ for the near optimal meshes (blue), while only two peaks centered in $l - l_{max} = -1$ and $l - l_{max} = 0$ exist for the two other meshes (red). Similarly, the normalized cell area distribution present one peak centered in $l - l_{max} = -2$ and $l - l_{max} = -5$ for, respectively, the $\frac{\eta_N}{\eta_{N,opt}} \approx 1$ meshes and the $\frac{\eta_N}{\eta_{N,opt}} \approx 0.3$ meshes. If we link this observation with the evolution of the loss factor, it seems that the error explosion



(a) 30 000 cells, $\frac{\eta_N}{\eta_{N,opt}} \approx 1$, $l_{max} = 9$, $L \approx 1.4$

(b) 30 000 cells, $\frac{\eta_N}{\eta_{N,opt}} \approx 0.3$, $l_{max} = 10$, $L \approx 19.8$



(c) 120 000 cells, $\frac{\eta_N}{\eta_{N,opt}} \approx 1$, $l_{max} = 10$, $L \approx 1.4$

(d) 120 000 cells, $\frac{\eta_N}{\eta_{N,opt}} \approx 0.3$, $l_{max} = 11$, $L \approx 11.7$

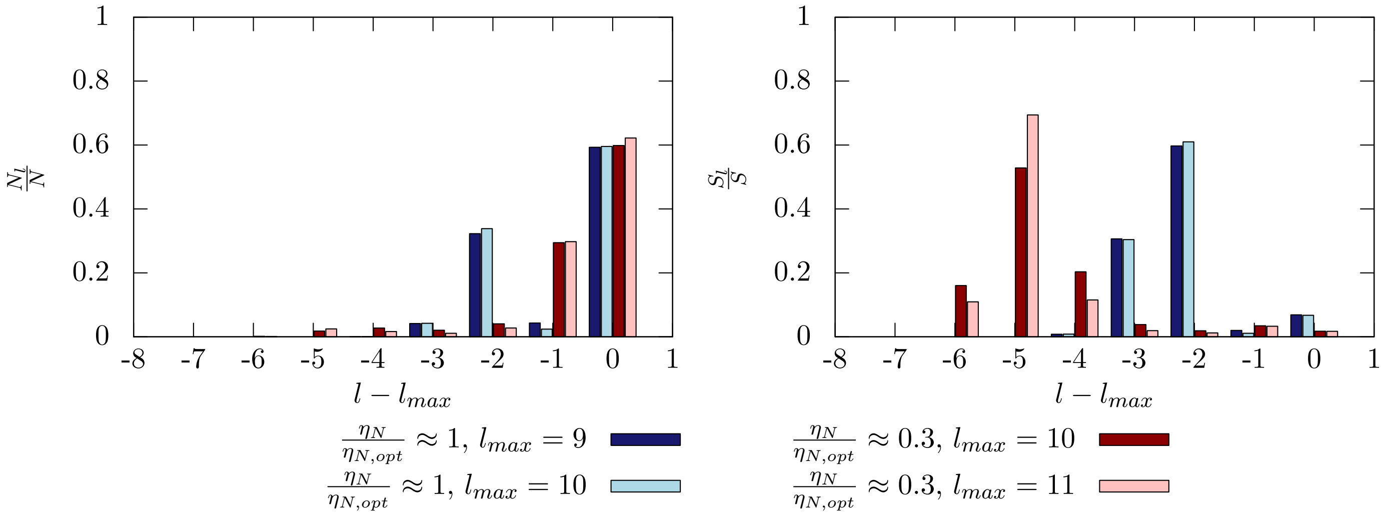
Figure 4.5: Meshes obtained for two compression ratio and for different minimum cell size.

is due to a too highly compressed mesh, with too coarse cell in the majority of the domain compared to too fine cells in a small region.

In table 4.1, we report the interpolation error – computed from the cell-centered analytical solution – and the total error – computed from the cell-centered numerical solution – measured in $\boldsymbol{L}^1$ and $\boldsymbol{L}^2$-norm for the meshes presented in figure 4.5. It is particularly interesting to remark that two different meshes with a similar number of grid points may have a comparable interpolation error but a very distinct total error, which is in agreement with the observations made in figure 4.4.

In $\boldsymbol{L}^1$ norm, for a given number of element, the interpolation error of the $\frac{\eta_N}{\eta_{N,opt}} \approx 0.3$ meshes is slightly lower than the interpolation error of the $\frac{\eta_N}{\eta_{N,opt}} \approx 1$ meshes. In $\boldsymbol{L}^2$ norm, the interpolation error is twice lower on the $\frac{\eta_N}{\eta_{N,opt}} \approx 0.3$ meshes than on the $\eta_N \approx 0.11$ meshes. However, for both norms, the total error is more than 10 times higher on the $\frac{\eta_N}{\eta_{N,opt}} \approx 0.3$ meshes than on the $\frac{\eta_N}{\eta_{N,opt}} \approx 1$ meshes.

To better understand this effect, we compare the local numerical error, measured using the numerical solution, and the exact local interpolation error obtained from the cell-centered analytical solution. In figure 4.7, we plot the local numerical error in function of the local



(a) Normalized cell size distribution.

(b) Normalized cell area distribution.

Figure 4.6: Normalized distributions for the meshes represented in figure 4.5. The distributions for the two $\frac{\eta_N}{\eta_{N,opt}} = 1$ meshes (light and dark blue) are identical. They are different from the distributions of the two $\frac{\eta_N}{\eta_{N,opt}} = 0.3$ meshes (light and dark red).

			$\boldsymbol{L}^1$		$\boldsymbol{L}^2$	
Mesh	N	$\frac{\eta_N}{\eta_{N,opt}}$	interpolation error	total error	interpolation error	total error
a	30 346	1	$1.6\ 10^{-4}$	$1.7\ 10^{-4}$	$1.0\ 10^{-3}$	$5.7\ 10^{-4}$
b	30 454	0.3	$1.5\ 10^{-4}$	$4.6\ 10^{-3}$	$5.0\ 10^{-4}$	$8.2\ 10^{-3}$
c	118 171	1	$3.9\ 10^{-5}$	$6.5\ 10^{-5}$	$2.5\ 10^{-4}$	$1.5\ 10^{-4}$
d	115 453	0.3	$3.7\ 10^{-5}$	$7.2\ 10^{-4}$	$1.2\ 10^{-4}$	$1.3\ 10^{-3}$

Table 4.1: Interpolation and total error comparison for the meshes represented in figure 4.1. Two different meshes may have a comparable interpolation error, but a huge gap between their total error.

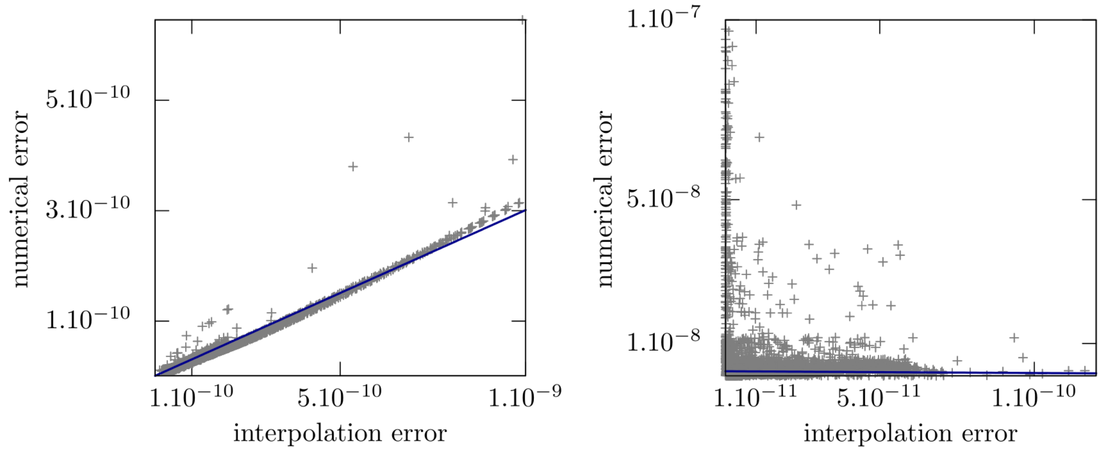
interpolation error for the four meshes presented in figure 4.1. While for the two meshes with a compression ratio $\frac{\eta_N}{\eta_{N,opt}} \approx 1$ ($L \approx 1.4$), we can see that there is a linear correlation between the two errors, they are completely decorrelated on the meshes with $\frac{\eta_N}{\eta_{N,opt}} \approx 0.3$ ($L > 10$). This observation generalizes well for the meshes obtained through the coarsening study, as we can see in figure 4.7e which shows the regression coefficient r^2 of the linear fits between both errors. In this figure, two main regimes are observable separated by a sharp intermediate regime. For compression ratios close to 1, a linear correlation is observed between the numerical error and the interpolation error and the regression coefficient r^2 is close to 1. A dramatic drop of r^2 occurs at $\frac{\eta_N}{\eta_{N,opt}} < 1$, where r^2 is almost null: the two errors are completely decorrelated. Moreover, in many elements of the mesh, the local numerical error is much higher than the local interpolation error. It is interesting to remark that the loss of linear correlation between the numerical and the interpolation errors occurs for a compression ratio η_N similar to the optimal compression ratio $\eta_{N,opt}$. Thus, the loss of linear correlation seems to be linked to excessively broad distributions of the grid size of the generated meshes.

As a conclusion, a sub-optimal grid from the point of view of a pure interpolation minimization problem may significantly reduce the total error of a numerical solution, resulting in an improvement of the efficiency of the adaptation method. Adding this observation to the previous ones, it follows that limiting the mesh compression could be an efficient way to avoid meshes where the solution contains large numerical errors. How to set the value for this limit remains to be clarified in the following sections.

Summary

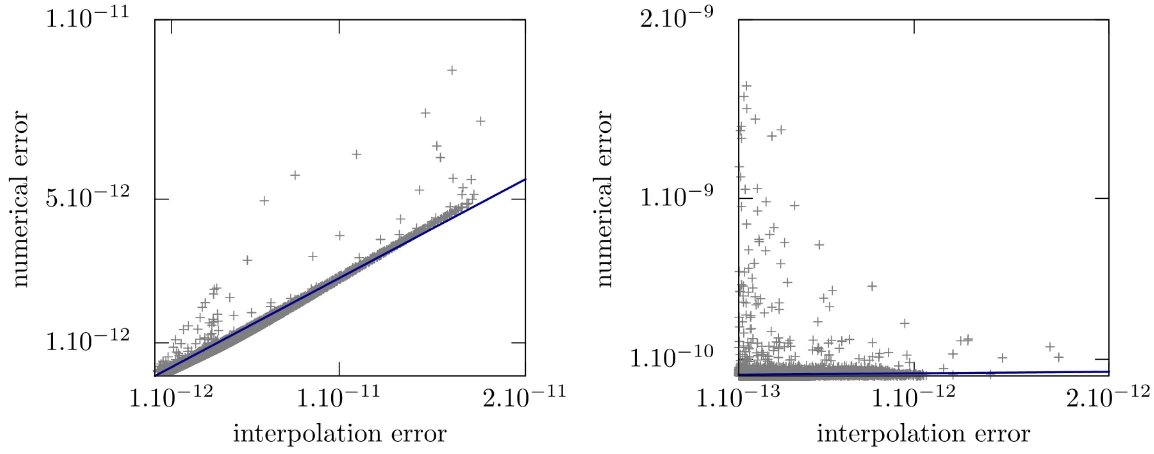
To conclude, through an interpolation error driven experiment, the numerical error may see several regimes when fixing the maximum level of refinement. The experiment presented is a coarsening study (an initial fine cartesian mesh is coarsen in regions of minimal interpolation errors), but the same regimes are observed when performing a refinement study (an initial coarse mesh is refined where the interpolation error is maximal), as shortly presented in appendix D. A first regime appears for meshes close to a uniform grid containing element with the minimum grid size, where the error does not depend on the number of elements. This regime is characterized by a linear correlation between the local numerical error and the local interpolation error and spans until the total error is close to the theoretical optimal interpolation error. An optimal compression ratio $\eta_{N,opt}$ is shown to exist for which the total numerical error is minimized. Coarsening further the grid implies a loss of correlation between the local interpolation errors and the total error in the numerical solution. The meshes of these regimes contain broad element distributions responsible for a dramatic increase of the error. In the context of mesh adaptation, the identification of regimes where the total error of a numerical solution observed is correlated with the interpolation error is compulsory for the direct application of local AMR strategies. The second regime should be avoided to obtain efficient local AMR methods. The regimes identified are independent from the level of refinement of the mesh and depend only on the compression ratio.

The optimal mesh that minimizes the total error is not necessarily the mesh which minimizes the interpolation error: a slightly sub-optimal towards the interpolation error, can be significantly more efficient to reduce the error in the numerical solution. The best AMR performances are expected for meshes with a compression ratio η_N as close as possible to the optimum $\eta_{N,opt}$, where the value of $\eta_{N,opt}$ is problem dependent and not known *a priori*. Thus, using the interpolation error measured from the numerical solution as an AMR criterion is not sufficient and the explicit addition of a supplementary constraint seems necessary.



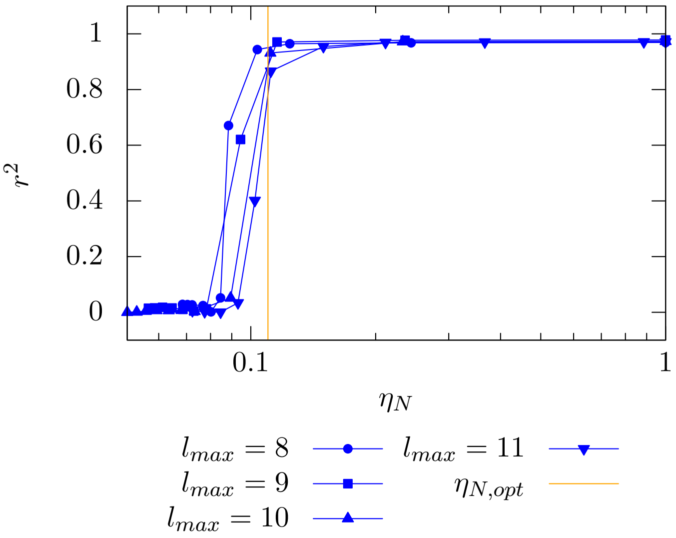
(a) $\frac{\eta_N}{\eta_{N,opt}} = 1$, $l_{max} = 9$

(b) $\frac{\eta_N}{\eta_{N,opt}} = 0.3$, $l_{max} = 10$



(c) $\frac{\eta_N}{\eta_{N,opt}} = 1$, $l_{max} = 10$

(d) $\frac{\eta_N}{\eta_{N,opt}} = 0.3$, $l_{max} = 11$



(e) Coefficient of determination r^2 obtained for the linear fit of the local numerical error in function of the local interpolation error.

Figure 4.7: Top and middle: Local numerical error *vs* local interpolation error for varying compression ratio η_N and maximum level of refinement l_{max}. Bottom: Coefficient of determination r^2 of the linear fit for all of the meshes of the coarsening study. For $\frac{\eta_N}{\eta_{N,opt}} \geq 1$ both errors are linearly correlated. For $\frac{\eta_N}{\eta_{N,opt}} < 1$, they are completely decorrelated.

2 A simple error model for PDE equations

In the previous section we have seen that the direct minimization of the interpolation error does not always guarantee the minimization of the numerical error. The direct explanation for this is the existence of regimes in which the total error is no longer well represented by a local representation of the error. It is today well accepted that the error may propagate from a region of production to other regions of the numerical domain, and it is often required to solve an adjoint problem or an error transport equation to take them into account. However, these approaches are computationally demanding and require to solve a new set of equations. In this section we explore the possibility to use the local interpolation error to develop a new error model that depends on the equation being solved.

Let Ω be a subset of $\mathbb{R}^n$ and Ω_h be the discretization of Ω.

Let u and u^h be the solution of a linear PDE on Ω and Ω_h.

The discretization of the linear PDE applied to u and u_h writes in matrix form:

$$\mathcal{A}u = b \ , \tag{4.7}$$

$$\mathcal{A}u_h = b_h \ . \tag{4.8}$$

$\mathcal{A}$, b and b_h depend on the equation to be solved and its discretization. Subtracting equation (4.7) to equation (4.8), we obtain the error equation, which follows

$$\mathcal{A}(u - u_h) = b - b_h \ . \tag{4.9}$$

Decomposing the matrix as $\mathcal{A} = \mathcal{D} + \mathcal{P}$, with $\mathcal{D}$ a diagonal matrix, we obtain

$$(\mathcal{D} + \mathcal{P})\,(u - u_h) = b - b_h \ . \tag{4.10}$$

The error equation solves as

$$(u - u_h) = \mathcal{D}^{-1}(b - b_h) - \mathcal{D}^{-1}\mathcal{P}(u - u_h) \ . \tag{4.11}$$

Assuming that the numerical error originates from the interpolation error and propagates only in a small neighbourhood, we modelize the right-hand side terms of equation (4.11) as interpolation errors to estimate the local numerical error:

$$|u - u_h| \leq |\mathcal{D}^{-1}|\,|b - \Pi_h b| + |\mathcal{D}^{-1}\mathcal{P}|\,|u - \Pi_h u| \ , \tag{4.12}$$

where the interpolation error in equation (4.12) are estimated using the isotropic Riemannian metric based optimal interpolation expression in equation (B.24).

This hypothesis is justified by the experiment presented in section 1, which shows on an example that a regime exist in which the error is dominated by the interpolation error. This is in agreement with classical results for finite element methods: the Céa's lemma [128] allows to bound the numerical error by the interpolation error in energy norm for elliptic problems [129, 125]. The practice suggests that this work even for non-elliptic problems [130]. This holds also for finite volume methods, where the truncation error mainly comes from the interpolation error of the fluxes [36]. Similarly, goal-oriented error estimates [33] for the Poisson-Helmholtz equation may write as a adjoint-weighted interpolation error.

For the Poisson-Helmholtz equation

$$\nabla \cdot (D\nabla u) + \lambda u = s \ , \tag{4.13}$$

the equation (4.7) writes using the matrix-like cell indexation as

$$\frac{D_{[1/2]}u_{[1]} - (D_{[-1/2]} + D_{[1/2]})u_{[0]} + D_{[-1/2]}u_{[-1]}}{h^2} + \lambda u_{[0]} = s_{[0]} \ , \tag{4.14}$$

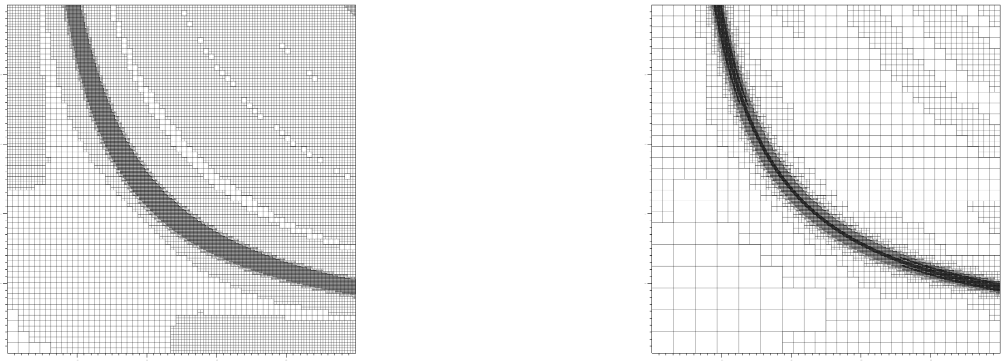
see chapter 2 section 2.1. The generalization to 2D and 3D is direct. Thus, estimation equation (4.12) reduces to

$$|u - u_h|_{[0]} \leq \frac{h^2|s - \Pi_h s|_{[0]} + D_{[1/2]}|u - \Pi_h u|_{[1]} + D_{[-1/2]}|u - \Pi_h u|_{[-1]}}{| - h^2\lambda_{[0]} + D_{[1/2]} + D_{[-1/2]}|} \ . \tag{4.15}$$

Finally, the discretized error estimate (4.12) that we obtained is a weighted mean in a small neighbourhood of the interpolation error of both the solution u and the source term b of the Poisson-Helmholtz equation. It accounts thus for a local error propagation.

3 Mesh compression ratio constrain

In section 1 of the current chapter, we concluded from an illustrative example that a too highly compressed mesh obtained by minimizing the interpolation error can produce grids with poor numerical solutions. These meshes are characterized by the presence of too coarse cells compared to the minimal cell sizes as we can see in figure 4.8.



(a) A nearly optimal grid ($\frac{\eta_N}{\eta_{N,opt}} \approx 1$) mesh with $N = 30\,000$ cells. Interpolation error: 1.6 10^{-4}. Total error: 1.7 10^{-4}.

(b) A too compressed ($\frac{\eta_N}{\eta_{N,opt}} \approx 0.3$) mesh with $N = 30\,000$ cells. Interpolation error: 1.5 10^{-4}. Total error: 4.6 10^{-3}.

Figure 4.8: Mesh compression comparison.

This issue is similar to mesh gradation issues already observed by other authors and reviewed in section 1.3 [80, 2]. In this section, we propose a new AMR method based on imposing an additional constrain on the compression ratio which is adapted to the Basilisk grid structure to limit the minimum element size h_{min} depending only on the sensor u and the objective number of elements N. Note that the complete adaptation process still depends on u and N only, as the explicitly imposed constrain $h_{min}(u, N)$ depends on u and N only. The estimation of the minimal cell size proceeds in two steps: first, a constrain on the compression ratio is proposed using two different methods proposed in sections 3.1.1 and 3.1.2. Then, this information is used to impose a minimal cell size as described hereafter.
Let Ω_h be a discretization of the domain Ω containing N elements. Let h_{min} be the minimal cell size of Ω_h that we want to obtain. First, we introduce $\Omega_{h_{min}}$, a uniform mesh constituted

by $N_{h_{min}}$ cells of size h_{min}. By definition, the cell size of a uniformly refined mesh is directly linked with the characteristic length L_0 and its number of elements:

$$h_{min} = \frac{L_0}{N_{h_{min}}^{\frac{1}{n}}} \ , \tag{4.16}$$

with n the problem dimension.

Introducing the compression ratio η_N of Ω_h as defined in equation (4.2), $\eta_N = \frac{N}{N_{h_{min}}}$, we readily obtain

$$h_{min} = L_0 \left(\frac{\eta_N}{N}\right)^{\frac{1}{n}} \ . \tag{4.17}$$

Given the compression ratio η_N, h_{min} is determined by the targeted number of grid points N. In the context of mesh adaptation, Ω_h is an adapted mesh, and N is prescribed by the user. Thus, the minimal element size is fully determined as long as η_N is imposed.

In Basilisk, the minimal cell size is equivalent to, and imposed as, the maximum mesh level l_{max} (equation (4.20)). By definition, we have

$$h_{min} = \frac{L_0}{2^{l_{max}}} \ . \tag{4.18}$$

From that, we obtain the maximum level as

$$l_{max} = \frac{\ln(L_0/h_{min})}{\ln(2)} \ . \tag{4.19}$$

Finally, introducing equation (4.17) and noting that the level l_{max} is an integer, we obtain

$$l_{max} = \left\lfloor \frac{\ln\left(\frac{N}{\eta_N}\right)}{n \ \ln(2)} \right\rfloor \tag{4.20}$$

where $\lfloor \cdot \rfloor$ denotes the floor value. The minimal cell size constraint will be imposed in the AMR method using an estimation of the optimal compression ratio, which is obtained in the next section.

3.1 Estimation methods for the compression ratio constrain

The minimization of the total error is characterized by an optimal compression ratio $\eta_{N,opt}$, which is *a priori* unknown. However, in section 1 of this chapter, we have seen that the total error remains roughly constant in the first regime, where the mesh is close to a uniform mesh. This is illustrated in figure 4.9: starting from a uniform mesh, we coarse some cells keeping a constant minimal element size and move along an almost horizontal line where the error increase is negligible. This example clearly shows that, by prescribing the minimum cell size, it is possible to generate meshes with nearly optimal performance. Based on this observation, we propose to compute h_{min} by imposing that this quantity is given by the element size of a uniform mesh that possesses the same interpolation error than the optimal mesh with N cells. This defines an additional constrain on the compression ratio reducing the size of the problem of optimization. In practice, imposing h_{min} implies that the the interpolation error is going to be larger than the optimal interpolation error value for a fixed N. The objective is to exclude the regions where we have observed that it is possible to obtain numerical errors

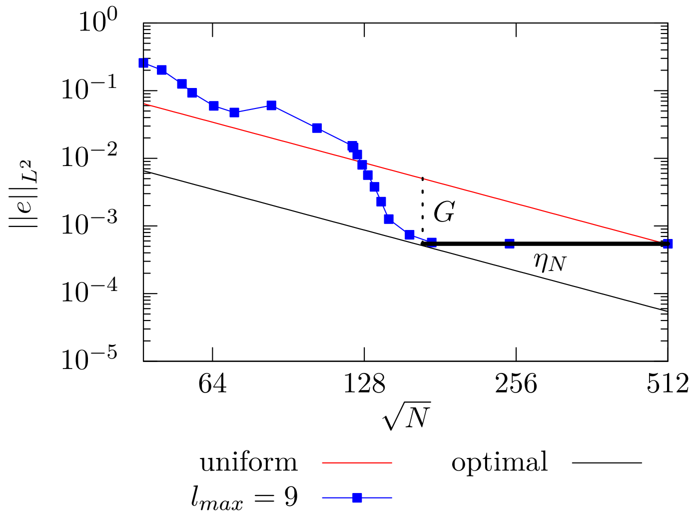


Figure 4.9: Representation of the approximated optimal compression ratio $\eta_{N,opt}$ and the associated gain factor G_{opt} on a mesh with the minimal cell size $h = \frac{1}{512}$. The numerical results come from the coarsening study in section 1.

that are orders of magnitude larger than the interpolation errors.

In the following, we propose two different methods to estimate this compression ratio. The first is a tree-based algorithm and the second is a metric-based theoretical estimation. Both methods are compared and discussed afterwards, and the interest to introduce a security coefficient is emphasized.

Before that, we recall in figure 4.10 some notations which will be used in the next sections. The optimal adapted mesh contains N elements. The uniformly refined meshes, only composed with elements having the size of the minimal size element of the adapted mesh, contains $N_{h_{min}}$ elements. The error $||e||_u$ on the uniform mesh approximates the error of the adapted mesh. A second fictive uniform mesh is characterized by its N elements and its error $||e||_{h_{u,N}}$.

3.1.1 A tree-based algorithm to estimate the compression ratio

The algorithm to estimate the compression ratio constrain η_{tree} is based on the Basilisk quad/octree grid structure. It is illustrated in figure 4.11, and summarized in the algorithm 4.

Let's consider a quad/octree mesh Ω_h and its tree structure $\mathcal{T}_h$. Ω_h contains only the leaf cells (*cf.* chapter 2, section 1) of $\mathcal{T}_h$. We obtain the solution u defined in all leaf cells K_{leaf} and noted u_{leaf}. The leaf cell levels are noted l_{leaf}. The non-leaf cells K_l of level l contain the interpolation $\Pi_l u$ of u. The space-filling tessellation of each non-leaf cell is composed by leaf cells.

First, u_{leaf} is used as a reference to estimate the interpolation error in all non-leaf cells.

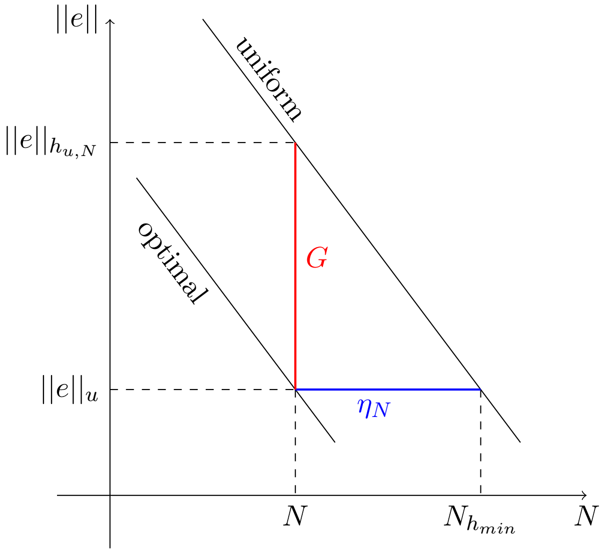


Figure 4.10: Schematic representation of the estimation of the optimal compression ratio $\eta_{N,opt}$ and the corresponding gain factor G.

In particular, the $\boldsymbol{L}^p$-norm error $||e||_{K_l}$ in a cell K_l of level l is estimated as

$$||e||^p_{K_l} = \int_{K_l} |u_{leaf} - \Pi_l u|^p(\boldsymbol{x})\ dK_l \tag{4.21}$$

$$\approx \sum_{K_{leaf} \in K_l} |u_{leaf} - \Pi_l u|^p(\boldsymbol{x}_{leaf})\ dK_{leaf} \tag{4.22}$$

Knowing the error in all non-leaf cells, the objective is to perform a *virtual* AMR study: fixing an ε criterion, a *virtual adapted mesh* $\mathcal{V}_h$ can be obtained by tagging the elements K_l respecting $||e||_{K_l} < \varepsilon$ and the quadtree structure constraint (two cells sharing a common direct parent cell must have the same level, and so the same tag), as illustrated in figures 4.12a and 4.12b. By convention, the smallest elements of $\mathcal{V}_h$ can't have a level greater than $l_{leaf} - 1$ [1]. Thus, the virtual mesh contains only non-leaf cells of $\mathcal{T}_h$. The tags of the virtual mesh cells are then updated, such that $\mathcal{V}_h$ meets the Basilisk mesh requirements (in particular the 1-irregularity constraint, *cf.* chapter 2) – see figure 4.12c. The tagged cells are used to count the number of element of the virtual mesh $N_{\mathcal{V}_h}$, and estimate its global interpolation error

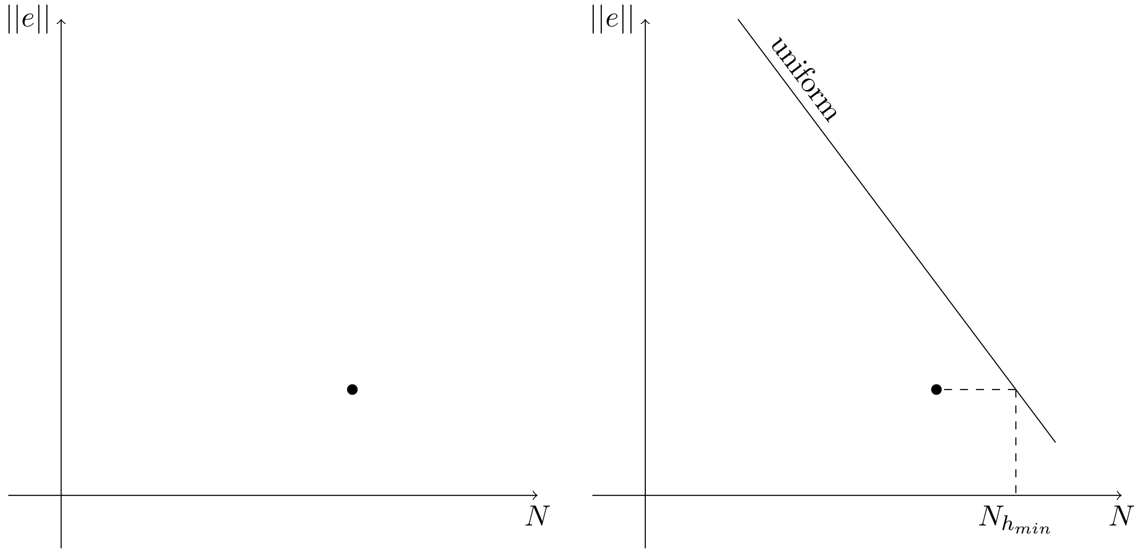
$$||e||^p_{\mathcal{V}_h} = \sum_{K_l \in \mathcal{V}_h} ||e||^p_{K_l}\ . \tag{4.23}$$

This operation is repeated for k_{max} increasingly small values of ε, thus we perform a *virtual coarsening study*. When $\varepsilon = 0$, the virtual mesh is constituted by the parents of the leaf cells. This virtual mesh is used to estimate the error $||e||_u$ and the number of elements $N_{h_{min}}$ defined previously.

For $\varepsilon \neq 0$, the compression ratio $\eta_{N,\mathcal{V}_h}$ and the gain factor $G_{\mathcal{V}_h}$ (*cf.* equation (4.5)) of the various virtual meshes are computed and stored. The gain factor compares the error on

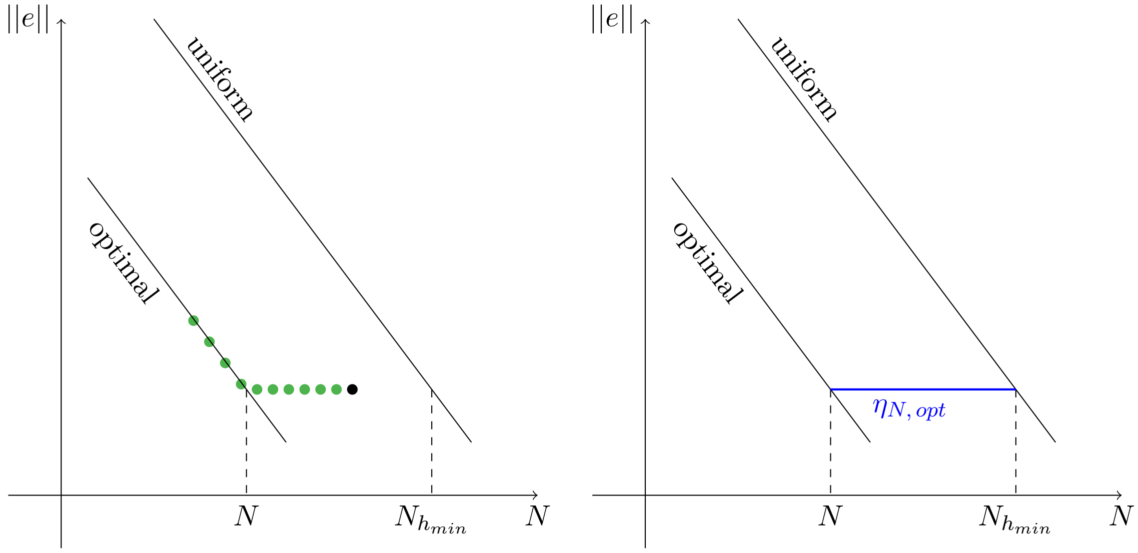
---

[1] No error can be computed in the leaves as they contain the reference solution.



(a) Initial mesh

(b) Estimation of the uniform error convergence curve ($k = 0$)



(c) *Virtual* coarsening study.

(d) Estimation of $\eta_{N,\,opt}$

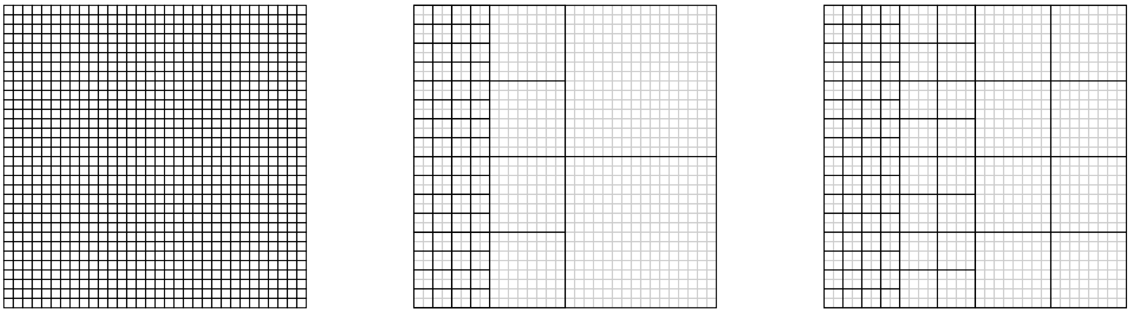
Figure 4.11: Steps to estimate $\eta_{N,\,opt}$ using the virtual AMR algorithm.

the virtual adapted mesh $||e||_{\mathcal{V}_h}$, and the error $||e||_{h_{u,N}}$ of the uniform mesh having the same number of elements that the adapted mesh, as illustrated in figure 4.10. In dimension n and assuming a second order convergence, $||e||_{h_{u,N}}$ expresses as

$$||e||_{h_{u,N}} = ||e||_u \frac{N_{h_{min}}^{2/n}}{N_{\mathcal{V}_h}^{2/n}} \tag{4.24}$$

From that, it readily comes

$$G_{\mathcal{V}_h} = \frac{||e||_{h_{u,N}}}{||e||_{\mathcal{V}_h}} = \frac{N_{h_{min}}^{2/n}||e||_u}{N_{\mathcal{V}_h}^{2/n}||e||_{\mathcal{V}_h}} \tag{4.25}$$



(a) Quadtree mesh Ω_h

(b) Virtual mesh $\mathcal{V}_h$ respecting an ε criterion and the quadtree structure.

(c) Corresponding virtual mesh $\mathcal{V}_h$ respecting Basilisk grid requirements

Figure 4.12: Example of virtual meshes respecting a given error criterion.

This loop is executed k_{max} times, which provides k_{max} values for the gain $G_{\mathcal{V}_h}$ and the compression ratio $\eta_{N,\mathcal{V}_h}$. Among them, we pick the compression ratio $\eta_{N,\mathcal{V}_h}$ corresponding to the maximum gain $G_{\mathcal{V}_h}$. Thus, we obtain an approximation of the optimal compression ratio η_{tree}. For each loop k, the ε criterion is set to $\varepsilon = 1.3||e||_{mean}(k/k_{max})^3$, with $||e||_{mean}$ the mean error of the cells at level $l = l_{leaf} - 2$. Thus, the value of ε varies from zero and the mean error evaluated on a mesh coarsen twice, which gives a wide range of virtual meshes.

Algorithm 4: Compression ratio estimation with *virtual* coarsening study.

```
Compute the interpolation error $||e||_{K_l}$ in all non-leaf cells;
Compute $||e||_{mean}$;
for $k$ in $[\![0, 1000]\!]$ do
    Compute $\varepsilon(k, ||e||_{mean})$ ;
    Tag all non-leaf cells respecting $||e||_{K_l} < \varepsilon$ and the quadtree structure;
    Update tagged cells to meet the 1-irregularity Basilisk mesh requirements;
    Compute and store $G$ and $\eta_N$ corresponding to the tagged cells;
end
Find the maximum gain $G$ and the corresponding optimal compression ratio $\eta_{N,opt}$ ;
```

3.1.2 A theoretical metric-based compression ratio estimation

It is also possible to propose a metric-based method to estimate the compression ratio constrain η_{metric}.

In chapter 3, we proved that the minimal interpolation error of a function u, whose Hessian matrix is H, writes

$$||u - \pi_{\mathcal{M}} u||_{\Omega,opt} = C_{opt} N^{-2/n} \tag{4.26}$$

with C_{opt} defined as:

$$C_{opt} = C_n \left(\int_{\Omega} \left(\mathrm{tr}(|H|(\boldsymbol{x})) \right)^{\frac{np}{2p+n}} \, \mathrm{d}\boldsymbol{x} \right)^{\frac{2p+n}{np}} , \tag{4.27}$$

and $C_2 = C_3 = 1/12$.

We also proved in equation (3.63) that the total interpolation error on a uniform mesh writes

$$||u - \pi_{\mathcal{M}} u||_{\Omega,u} = C_u N^{-2/n} \tag{4.28}$$

with

$$C_u = C_n L_0^2 \left(\int_\Omega (tr(|H|)(\boldsymbol{x}))^p d\boldsymbol{x} \right)^{1/p} , \tag{4.29}$$

and $C_2 = C_3 = 1/12$.

If we assume that there exists a transition between the optimal grid and a uniform grid with the same minimum grid size for which the error remains constant, the optimal compression ratio is given as:

$$\eta_{metric} = \frac{N_{opt}}{N_{h_{min}}} = \frac{C_{opt}\ ||u - \pi_{\mathcal{M}} u||_{\Omega,\,u}}{C_u\ ||u - \pi_{\mathcal{M}} u||_{\Omega,\,opt}} \ . \tag{4.30}$$

Moreover, this quantity is estimated at iso-error, as illustrated in figure 4.10. Thus, the compression ratio reduces to:

$$\eta_{metric} = \frac{C_{opt}}{C_u} \ . \tag{4.31}$$

It is particularly interesting to remark that this quantity can be obtained analytically for any given function. It only depends on the Hessian of the function, the dimension of the problem and the error norm. This additional restriction on the compression ratio can be considered as an intrinsic property of a sensor u.

3.1.3 Comparison between compression ratio estimation methods

Here, we compare qualitatively and quantitatively both compression ratio restrictions.

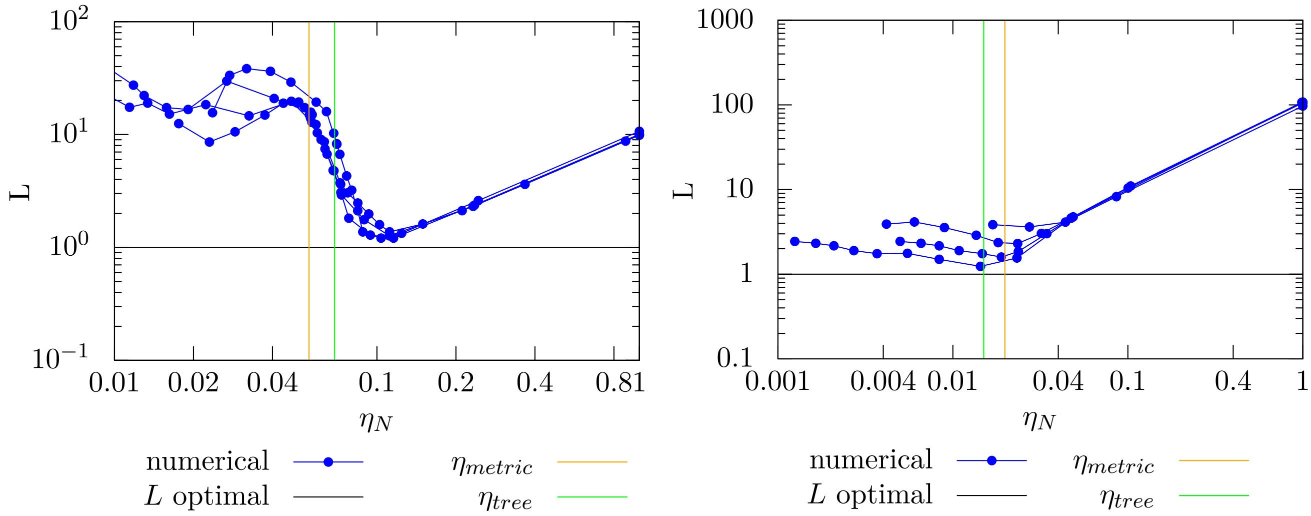
Qualitative comparison

On a qualitative point of view, both estimations are different. In particular, the algorithm in section 3.1.1 is naturally handled by the quad/octree grid structure. It uses all the information known about the sensor in the whole tree grid structure. However, its current implementation is not evident to parallelize for MPI. This is due to a mix between runs of the tree structure from the leaves to the level 0 cell – to estimate the interpolation error in all non-leaf cell – and runs from the level 0 cell to the leaves – to obtain a virtual mesh respecting the Basilisk mesh requirements. Through our current algorithm implementation, the same cell is not treated by the same MPI-process in both cases.

The theoretical estimation proposed in section 3.1.2 is based on the continuous mesh framework and shows that the optimal compression ratio deduces directly from the sensor. Moreover, as the same framework is used as a basis to estimate the error of a numerical solution, the necessary quantities does not need to be computed again, even if, intrinsically, they require higher resources to be computed (necessity to reconstruct the Hessian of the sensor). Finally, this estimation requires a minimal effort to be MPI-parallelized, as it only uses leaf cells traversal, which is an operation already parallelized in Basilisk.

Quantitative comparison

To estimate the quantitative performance of both methods, we compare the optimal compression ratio predictions η_{tree} and η_{metric} obtained respectively with the tree-based algorithm and the metric-based theory. This comparison is made using the results of a coarsening study made on the multiscale sinusoidal and the boundary layer cell-centered solutions presented in appendix C.



(a) Coarsening study based on the multiscale sinusoidal numerical solution (appendix C, equation (C.1))

(b) Coarsening study based on the boudary layer numerical solution (appendix C, equation (C.3))

Figure 4.13: Comparison between the optimal compression ratio provided by the metric-based theory η_{metric} and the tree-based algorithm η_{tree}. They are compared with the loss factor *vs* compression ratio curves obtained from coarsening studies based on the numerical solution (see section 1).

Figure 4.13 illustrates the results issued from these coarsening studies. We observe that both algorithms provide similar compression ratio. For the case of the multiscale sinusoidal function, they are both slightly underestimated, while they are slightly overestimated for the boundary layer solution. In both cases, η_{tree} seems to be a slightly better approximation of $\eta_{N,opt}$. The reason why both predictions are different is that the estimation of η_{metric} uses the additional iso-error hypothesis. Nonetheless, η_{metric} seems still to be a reliable estimation of $\eta_{N,opt}$ with a lower computational cost. This validates both optimal compression ratio predictions.

For the rest of this chapter, we present results obtained with the metric-based optimal ratio estimation. Some of them have also been run with the tree-based estimation without real difference in terms of performance. However, the metric-based estimation is attractive for MPI-parallelization and allows to perform finer mesh adaptations than the tree-based one.

3.2 Security coefficient

The previous sections provide two methods to compute an estimate of the compression ratio. However, when applied to real cases we have seen that this value can be smaller than the optimal compression ratio resulting in meshes with large numerical errors as detailed in section 1. The reason for this underestimation is that the point of optimal numerical performance is unknown. We have already shown that being too ambitious trying to reach an optimal mesh can lead to total errors that can be even larger than the ones obtained for cartesian meshes, which alter heavily the performance of the proposed method. This is a consequence of the loss of correlation between the interpolation and the total numerical error. One solution is to restrict the size of the mesh optimization problem to a subset of meshes with errors above a given threshold, as we have seen that as the mesh approaches a uniformly refined mesh, both interpolation and total numerical errors are correlated. Relaxing the objective of minimizing the error has two main advantages: first, the minimization problem size is reduced as we are going to exclude the meshes with interpolation errors below a given threshold. Secondly, we no longer look for a unique optimal mesh, but for a family of grids that reach a given error threshold. The existence of multiple suboptimal solutions in terms of interpolation error significantly increases the chances of success of a method in a real case.

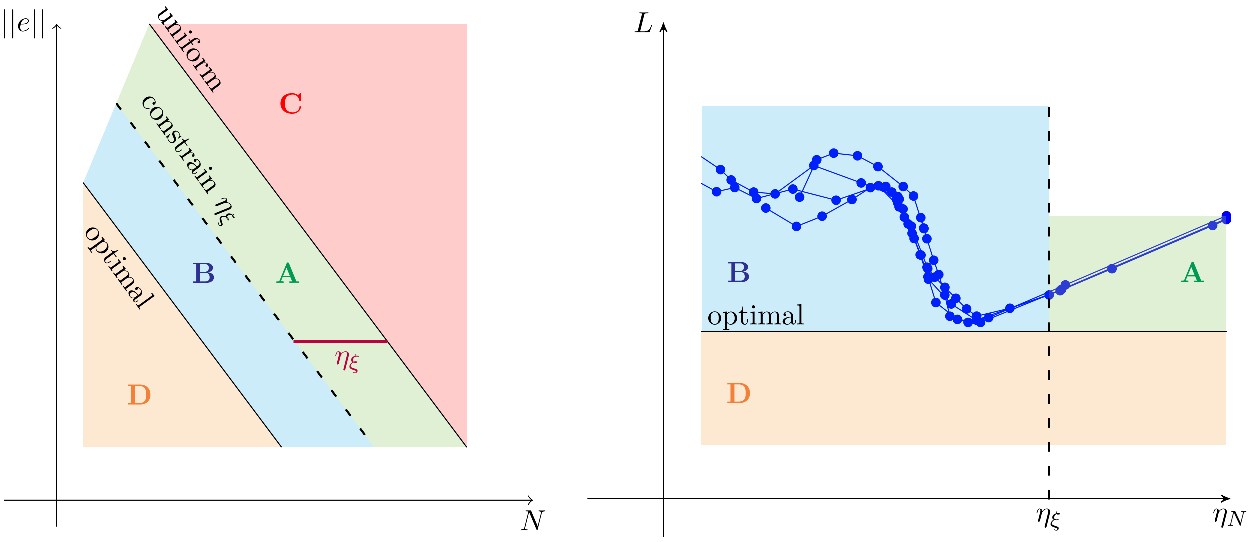
This effect is illustrated in figure 4.14, where all possible adapted meshes containing N elements are classified into three different groups (A,B,C). No meshes exist in region D as, by definition, there is not any mesh that can give $L < 1$. The region C holds for meshes which have detrimental performances in comparison with a series of uniformly refined meshes for pure interpolation errors. These grids are excluded by any local AMR algorithm based on the interpolation error. The regions A and B contain efficient adapted meshes in terms of pure interpolation error. The larger the compression ratio, the larger the targeted errors and the smaller the region of optimization (A), increasing the chances of success of purely local AMR methods at expenses of penalizing the minimum achievable error. Increasing the size of the optimization problem include meshes with potential smaller errors at the risk of loosing the correlation between local and global errors and therefore failing to minimize the total error.

Let's now introduce a security coefficient $\xi > 0$ and define the objective compression ratio η_ξ such that

$$\eta_\xi = (1+\xi)\eta_{N,obj} \ . \tag{4.32}$$

This objective compression ratio is strictly greater than the estimated optimal compression ratio in terms of pure interpolation error. The limit $\xi = 0$ implies that no security coefficient is imposed and that we are going to try to find a global optimal grid in terms of pure interpolation error. In practice, we use this objective compression ratio instead of the optimal one to estimate the minimal cell size in equation (4.17). As long as ξ is small, the result of the optimization problem should only result in a slightly under-performing AMR procedure towards the interpolation error (see figure 4.14) reducing the chances to enter in the regime of numerical error explosion.

The introduction of the security coefficient is then a parameter to control, in an error *vs* N diagram, the distance between the point representing the target grid of N elements and the cartesian grid where the simulation domain is uniformly refined with the minimum grid size. The larger its value, the more reliable the solution of the optimization problem is at expenses of penalizing the total error of the solution.



(a) Meshes representation in terms of the interpolation error convergence curves.

(b) Meshes representation depending on the loss factor L and the compression ratio η. The blue data correspond to the total error results presented in chapter 4, section 1

Figure 4.14: Schematic representation of the influence of a compression ratio constrain η_ξ. The different colored regions correspond to different efficiency of the meshes. (C): non-efficient adapted mesh in terms of interpolation error. (D): no mesh. (A) and (B): adapted meshes which reduce the interpolation error in comparison with uniformly refined meshes. Meshes in (A) respect $\eta \geq \eta_N$. The total error in (B) can be larger than the total error on a cartesian mesh.

4 Proposed AMR strategy

4.1 AMR algorithm

The global adaptive procedure – *cf.* algorithm 5 – is similar to the adaptive procedure described in chapter 3, section 4.3: it is performed iteratively until an objective number of elements is reached. However, an additional step is performed: the optimal compression ratio is estimated in order to compute automatically and impose a maximum level of refinement.

The adaptive algorithm proceeds as follow: first, the error is computed in each cell using the error model developed in section 2, *cf.* equation (4.12). Then, the compression ratio is evaluated and provide a maximum level as defined in equation (4.20). Once this is done, the cells are refined or coarsen by maximum one level, depending on the comparison between their error and an ε criterion. The maximum level constraint is explicitly enforced. Finally, the cells are counted and compared to the objective cell number. If it is not reached, ε is updated and the procedure is repeated until the desired number of grid points is reached.

Algorithm 5: Adaptation loop with compression ratio estimation.

```
Set the initial mesh $\mathcal{M}_0$ (coarse uniform Cartesian mesh);
Set the objective number of elements $N_{obj}$;
Set the initial $\varepsilon$;
while $N \neq N_{obj} \pm 10\%$ do
    Compute the solution $S_i$ on $\mathcal{M}_i$ ;
    Estimate the optimal compression ratio $\eta_{N,\,opt}$;
    Compute the maximum level of refinement $l_{max}(\eta_{N,\,opt})$;
    Compute the error $||e||_{l,\boldsymbol{L^p}}$ in each element;
    Restrict it to estimate the error $||e||_{l-1,\boldsymbol{L^p}}$ in the parent cell;
    foreach element do
        if $||e||_{l,\boldsymbol{L^p}} > \varepsilon$ and $l \leq l_{max}$ then
            Refine the cell;
        else
            if $||e||_{l-1,\boldsymbol{L^p}} < \varepsilon/4$ and neighboring cells are not too fine then
                Coarsen the cell;
            end
        end
    end
    Count the number of elements $N$ and update $\varepsilon$;
end
```

4.2 Illustrative example

As a validation example, we compare the AMR algorithm proposed here – which estimate the objective compression ratio to limit the maximum cell level $l_{max}(\eta_\xi)$ – with the same algorithm without the maximum cell level limit. The equation solved is

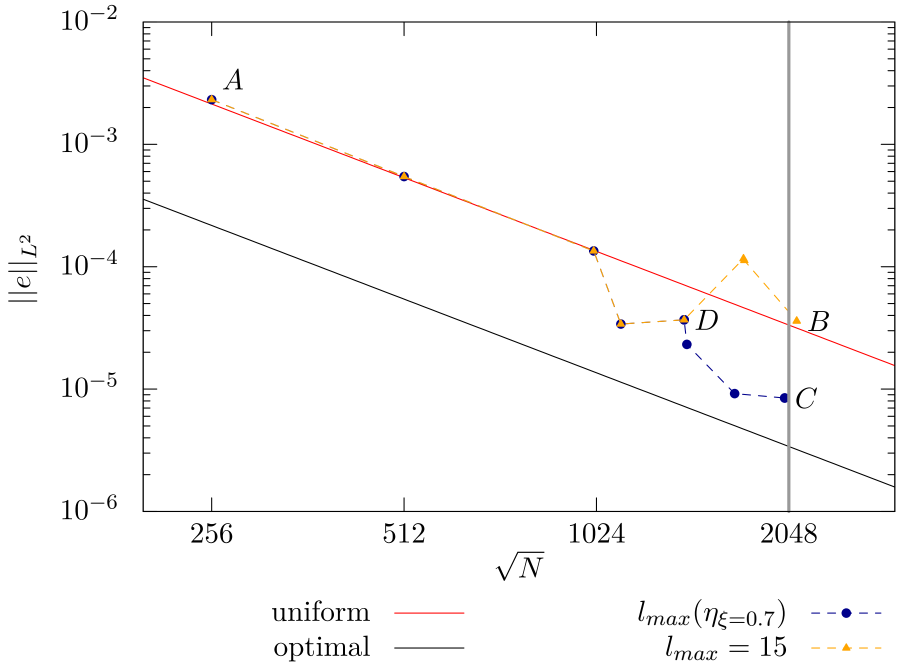
$$\nabla \cdot (D \nabla u) - u = s \ , \tag{4.33}$$

with $D = 10^{-2}$ and the source term s is computed such that the solution u is the multiscale sinusoidal solution (*cf.* appendix C, equation (C.1)).

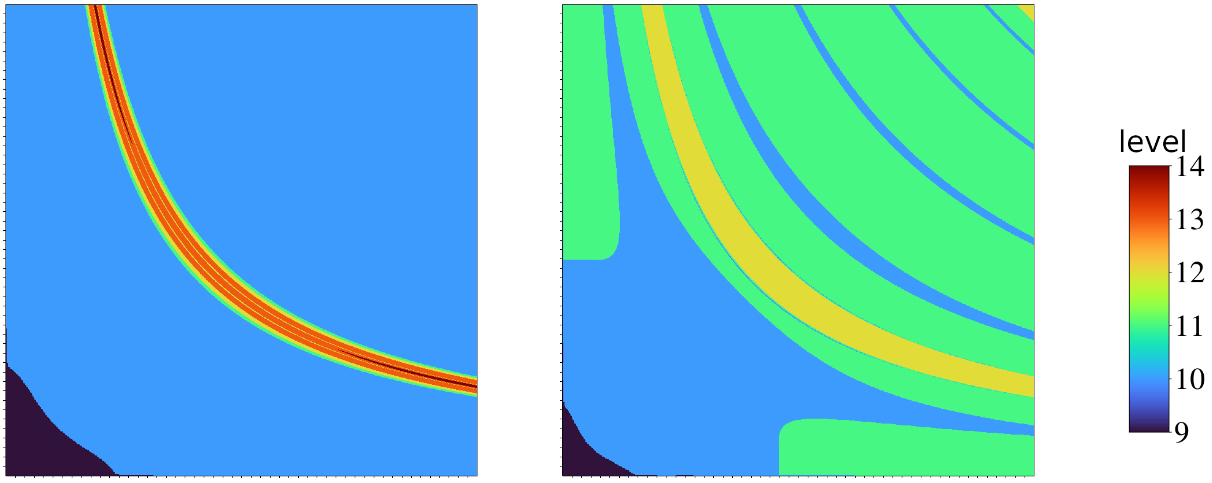
Starting from an initial uniform mesh A containing 65 536 elements, the objective is to obtain an adapted mesh containing 4 194 304 elements. The security coefficient is arbitrarily set to $\xi = 0.7$. At the end of the iterative algorithm, the mesh B is obtained for the case with $l_{max} = 15$, and the mesh C for the case $l_{max}(\eta_{\xi=0.7})$. At each iteration, the error on the intermediate solution is registered – see figure 4.15a – and the final mesh levels are shown in figure 4.15.

From figure 4.15a, we see that the first iterations of the algorithm are identical in both cases. This occurs as long as the whole mesh needs to be refined, and the maximum level l_i of the mesh of the iteration i respects $l_i <= l_{max}(\eta_\xi)$. This holds until the iteration noted D in the figure. After this iteration, l_i continue to increase in the case $l_{max} = 15$, leading to the mesh B, whereas $l_i = l_{max}(\eta_\xi)$ when the optimal compression ratio is enforced, leading to mesh C. The error on the solution on the mesh B is greater than the error obtained on a uniform mesh, whereas the error for the mesh C is lower than the uniform mesh error and closer to the theoretical minimal interpolation error than to the uniform mesh error. The mesh C is thus more efficient than the mesh B in terms of error reduction.

It is clear from figure 4.15 that the finest elements of mesh B are finer than the finest elements of mesh C, and that the majority of the mesh B is filled with bigger elements than in the mesh C. In other words, the compression ratio of the mesh B, $\eta_B \approx 0.017$, is lower than the compression ratio of the mesh C, $\eta_C \approx 0.24$, and we have the double inequality $\eta_B < \eta_\xi < \eta_C$. This test case illustrates the efficiency of the AMR method based on the optimal compression ratio estimation.



(a) The error at each iteration is shown and compared for the two algorithms ($l_{max}(\eta_{\xi=0.7})$ and $l_{max} = 15$).



(b) Mesh level B ($l_{max} = 15$). $\eta_B \approx 0.017$. $N = 4\,437\,349$ elements. Total error: $||e||_{\boldsymbol{L}^2,num} \approx 3.6\ 10^{-5}$. Interpolation error: $||e||_{\boldsymbol{L}^2,interp} \approx 3.4\ 10^{-6}$.

(c) Mesh level C ($l_{max}(\eta_{\xi=0.7})$). $\eta_C \approx 0.24$. $N = 4\,070\,809$ elements. Total error: $||e||_{\boldsymbol{L}^2,num} \approx 8.5\ 10^{-6}$. Interpolation error: $||e||_{\boldsymbol{L}^2,interp} \approx 1.6\ 10^{-5}$.

Figure 4.15: Result of the mesh optimization algorithm that minimises the interpolation error (bottom left) and the grid obtained when imposing the constrain on the minimum grid size (bottom right). The corresponding total error convergence is presented (top).

5 Validation

In this section, we present two validation test cases of our AMR method. The equation

$$\nabla \cdot (D\nabla u) - u = s \ , \tag{4.34}$$

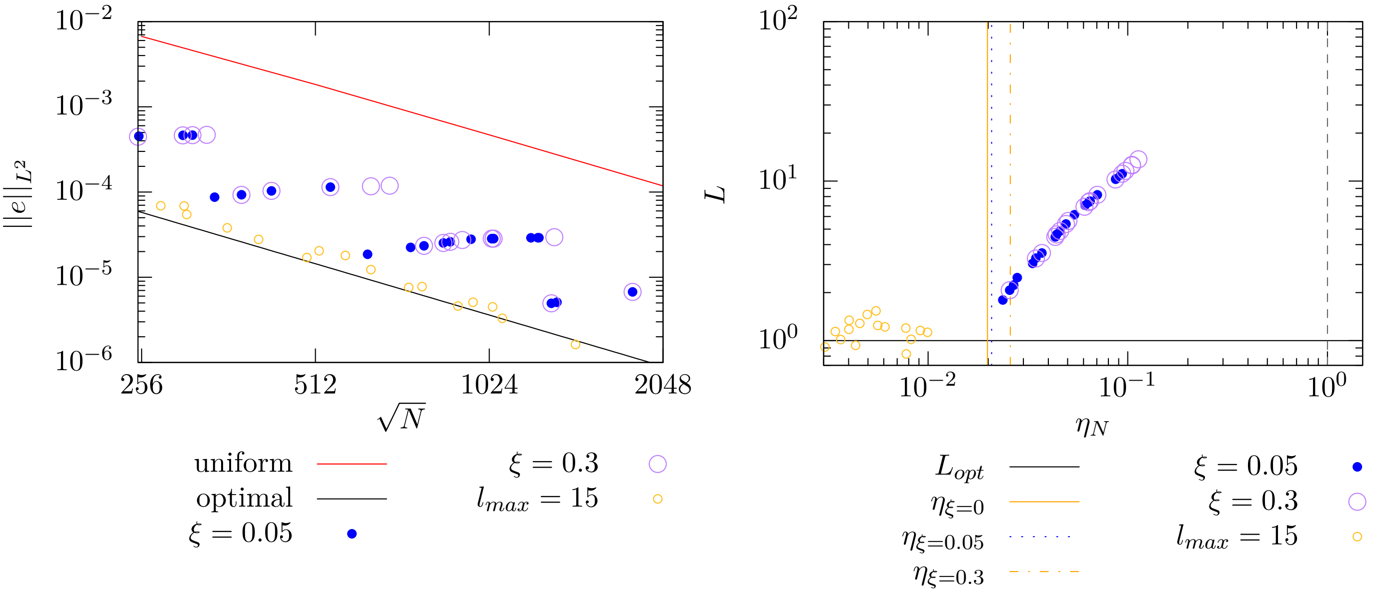
with $D = 10^{-2}$, is solved, and an error convergence study is performed: a series of adapted meshes are obtained and the error of the numerical solution is measured on each adapted mesh. The performance of our new AMR method is then compared with other AMR methods and assessed by comparison with the expected optimal error.

5.1 Boundary layer

In this first test case, we show the performances of our AMR method for a boundary layer problem defined through the solution

$$u_2(x, y) = xy^2 - y^2 \exp(\frac{2(x-1)}{\kappa}) - x \exp(\frac{3(y-1)}{\kappa}) + \exp(\frac{2(x-1) + 3(y-1)}{\kappa}) \ , \qquad (4.35)$$

with $\kappa = 10^{-2}$, described in appendix C. This problem contains only one scale controlled by the coefficient κ.



(a) L^2 norm error error convergence curves.

(b) Loss factor L *vs* compression ratio η_N representation.

Figure 4.16: AMR method performance evaluation for the solution (equation (C.3)) of the Poisson-Helmholtz equation (4.34). The current AMR method performs closely to the theoretical optimum with a security coefficient $\xi = 0.05$, and less closely to the optimal with $\xi = 0.3$. The AMR without compression ratio ($l_{max} = 15$) is near optimal.

In figure 4.16a, we can see the results obtained with our AMR method with two different security coefficients ($\xi = 0.05$ and $\xi = 0.3$) and the results without any compression ratio estimation ($l_{max} = 15$). For reference we also show the expected theoretical optimal error and the error on uniformly refined meshes. Three interesting points can be emphasized. First, the solution error on the adapted meshes (blue dots) is closer to the optimal curve with the security coefficient $\xi = 0.05$ than with $\xi = 0.3$. This difference of behavior is expected as a higher security coefficient implies that the subspace of meshes that we authorize as possible candidates to minimize the numerical error have poorer performances than the optimal mesh that minimizes the interpolation error only and that serves as reference.

Secondly, the error curve obtained with our AMR algorithm has a stair-like shape with a series of plateaus. This feature is due to the discrete minimal cell size prescription: the element size in Basilisk depends on the level of the element (see equation (2.1)) and is, by definition, an integer. Thus, it is not possible to gradually decrease its value that leads to sudden jumps every time the minimum grid size changes.

Finally, the results obtained without the compression ratio based automatic minimum cell size prescription ($l_{max} = 15$) are close to the optimal performance. This implies that, for this particular case, the interpolation error based AMR criterion still performs well to describe the total numerical error for near optimal meshes: no error explosion is observed, and therefore introducing an additional constrain leads to poorer performance of the method. However, this loss of performance is limited.

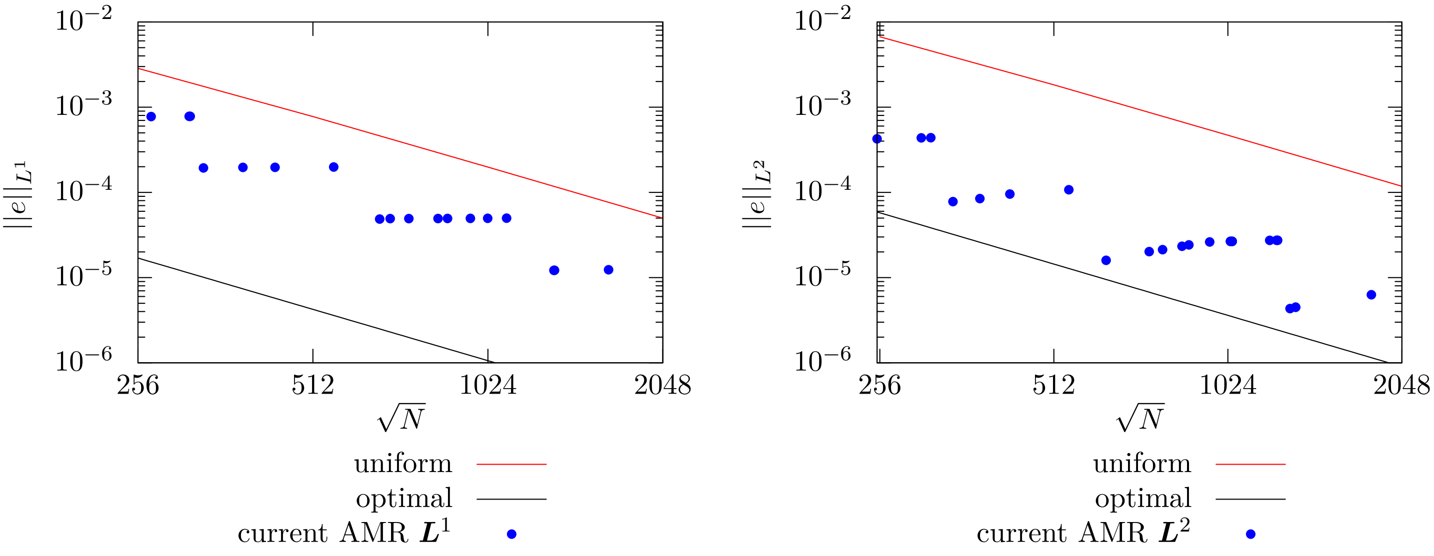
The results can be represented as the loss coefficient in function of the compression ratio. It is clear from figure 4.16b that our algorithm performs correctly, as the loss approaches the expected optimum $L = 1$. Moreover, the compression ratio of all adapted meshes is higher than the estimated optimal compression ratio modified by the security coefficient. Thus, the security coefficient is effective to control the mesh compression constrain, and the finally imposed minimal compression ratio is respected. This implies that the adapted meshes are all sub-optimal towards the direct interpolation error minimisation, but it ensures that the numerical error is controlled.

Discussion regarding the choice of error norm

Nowadays, it is commonly admitted that to reduce the error on a numerical solution using interpolation error-based criteria, at least the $\boldsymbol{L}^2$-norm should be used [12, 30, 114]. It is rather intuitive on a mathematical point of view – at least for finite-element methods – as the $\boldsymbol{L}^2$-norm error is naturally induced by variational formulations [59]. Moreover, numerical evidences show the pertinence of this error-norm for numerous flow fields, for example for flows with strong variations [30, 113]. On the opposite, $\boldsymbol{L}^1$-norms appear mainly when dealing with adjoint-based error estimations [35, 33]. However, goal-oriented methods consider an error measured on an output-functional rather than on the error on the complete solution.

In the previous sections, we performed AMR studies based on the $\boldsymbol{L}^2$-norm interpolation error. We can perform them in $\boldsymbol{L}^1$-norm to verify the performances of this error norm in combination with the Basilisk solver.

Figure 4.17 compares the AMR performed using $\boldsymbol{L}^1$ and $\boldsymbol{L}^2$ norms. As previously seen, the $\boldsymbol{L}^2$-norm error control in AMR performs closely to the expected optimum. However, the $\boldsymbol{L}^1$-norm optimal curve is not reached by the AMR method: a gap of around one decade exist between the optimal and the AMR errors. As the $\boldsymbol{L}^1$-norm interpolation error has been verified on analytical examples in chapter 3, section 5, these results show that the $\boldsymbol{L}^1$-norm AMR method performs less well to reduce the $\boldsymbol{L}^1$-norm of the numerical error than the $\boldsymbol{L}^2$-norm to reduce the $\boldsymbol{L}^2$-norm error. Thus, better AMR performance can be expected by choosing the $\boldsymbol{L}^2$-norm to perform mesh adaptation when dealing with numerical solutions of elliptic equations. This is in agreement with the general opinion.



(a) L^1 error

(b) L^2 error

Figure 4.17: Comparison between $\boldsymbol{L}^1$ and $\boldsymbol{L}^2$ AMR for the numerical boundary layer solution (equation (C.3)).

5.2 Multiscale sinus

For the second test case, the solution is the previously described multiscale sinusoidal function (appendix C, equation (C.1)):

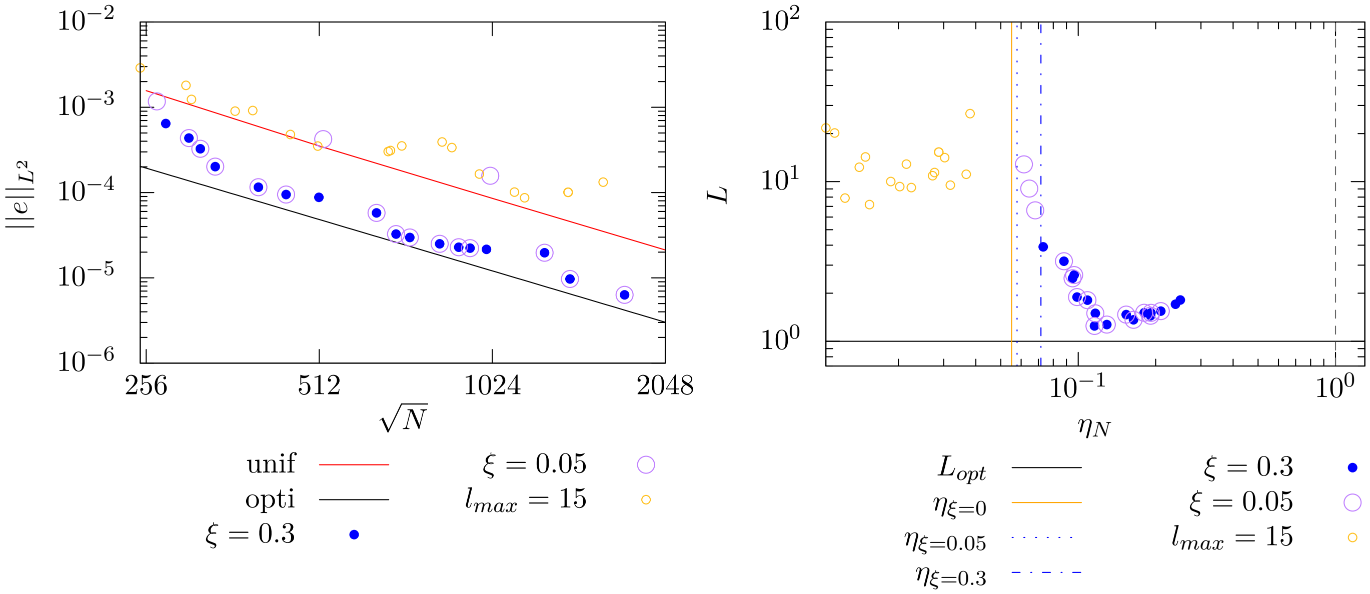
$$u_1(x, y) = \sin(a\,(xy - b))\left(0.01 + e^{-(d\,(xy-c))^2}\right) \tag{4.36}$$

with $a = 5\pi$, $b = 1.5\pi/50$, $c = 3\pi/50$ and $d = 50\pi$.

The figure 4.18a presents the error convergence results for the first test case. It compares the results obtained using the proposed AMR algorithm with two security coefficients $\xi = 0.3$ and $\xi = 0.05$ (and the associated maximum cell levels $l_{max}(\eta_{\xi=0.05})$ and $l_{max}(\eta_{\xi=0.3})$), and the same algorithm without optimal compression ratio estimation ($l_{max} = 15$). The minimal interpolation error and the uniform mesh error are shown as a reference.

Without the explicit optimal ratio constraint (empty blue circles), the AMR meshes produce a solution containing more error than a uniform mesh containing the same number of element, which means that the method performs poorly and is even detrimental toward the solution accuracy by comparison with a set of uniformly refined meshes. On the other hand, the results obtained with the algorithm which explicitly imposes the optimal ratio constraint with a security coefficient $\xi = 0.3$ are close to the theoretical optimal error. The results with a security coefficient $\xi = 0.05$ are intermediate between the two others: they are partially near optimal, and show partially less good performance than uniformly refined meshes.

Figure 4.18b represents the loss factor as a function of the compression ratio. It is clear that the algorithm based on the optimal compression ratio estimation provides adapted meshes close to the optimum and with a loss factor reduced in comparison with the algorithm with $l_{max} = 15$. This illustrates the interest of imposing explicitly the optimal compression



(a) L^2 norm error error convergence curves.

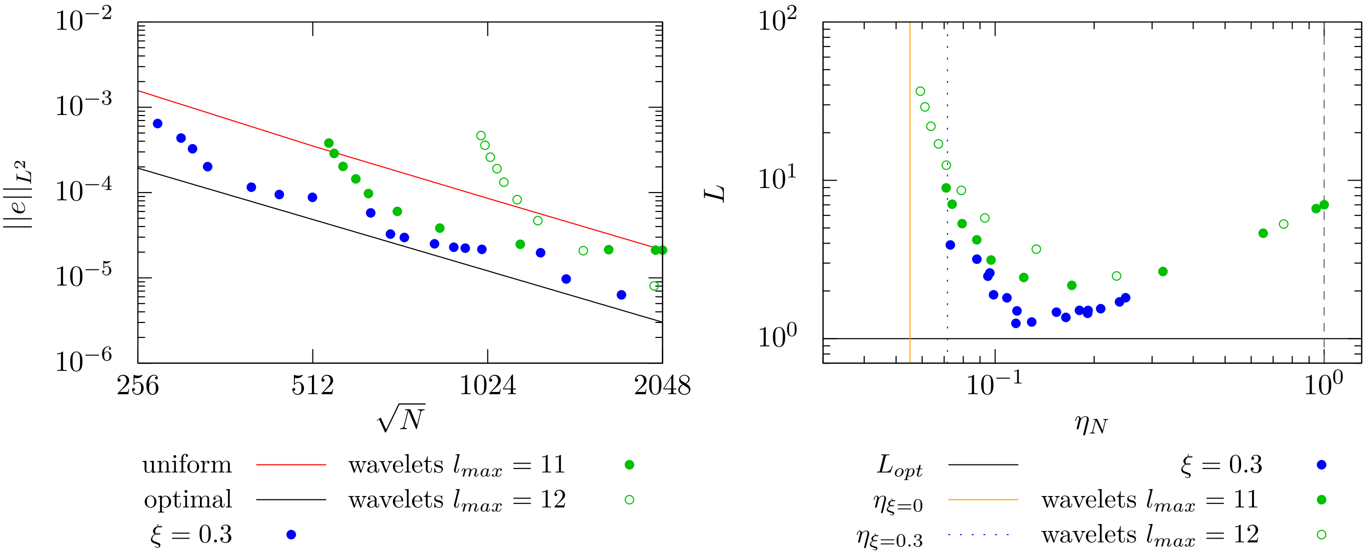
(b) Loss factor L *vs* compression ratio η_N representation.

Figure 4.18: AMR method performance evaluation for the solution (equation (C.1)) of the Poisson-Helmholtz equation (4.34) for two AMR methods: one imposing $l_{max} = 15$ (empty blue dots), the other based on the optimal compression ratio and imposing $l_{max}(\eta_{\xi=0.3})$ (filled blue dots).

ratio.

In figure 4.19a and 4.19b we compare the AMR performances of our new AMR algorithm with the wavelet-based AMR method of Basilisk. First, the wavelet-based method is sensible to the user-imposed minimal cell size, as we can see by comparing the two wavelet-based AMR studies performed with $l_{max} = 11$ (green dots) and $l_{max} = 12$ (green circles). This is problematic, especially if one sees that reducing the minimum grid size does not always guarantee to reduce the overall error as expected, it could lead to a significant increase of the numerical error contained in the solution. Here, for $\sqrt{N} \approx 1024$, the wavelet-based method gives a lower numerical error with $l_{max} = 11$ than with $l_{max} = 12$. As explained previously, the proposed method automatically provides the correct minimal cell size, which prevents the user to explicitely set any minimum grid size.

In addition, for any number of element N, the solution error is lower on the meshes obtained with the new proposed method than with the wavelet-based method. This is especially visible in figure 4.19b, where the new method provides meshes with a near optimal loss factor, whereas the wavelet-based algorithm is not able to achieve such good performance. Thus, our new method upgrades the existing AMR method of Basilisk in terms of error performance.



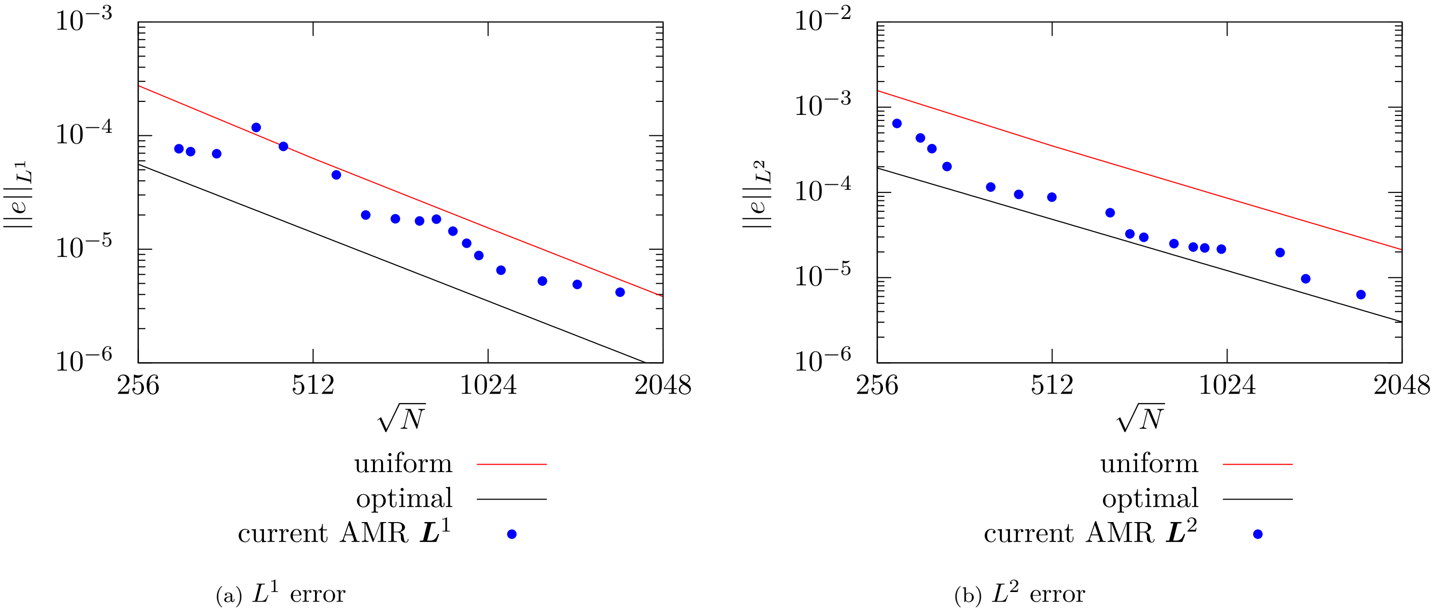
(a) L^2 norm error error convergence curves.

(b) Loss factor L *vs* compression ratio η_N representation.

Figure 4.19: Performances comparison between the current AMR method (blue dots) and the wavelet-based method of Basilisk (green dots and circles) for the multiscale sinusoidal numerical solution. The wavelet-based method is sensitive to the authorized maximum level. The current AMR method performs better (less error and loss factor for a given number of elements).

Discussion regarding the choice of error norm

Figure 4.20 compares the error convergences in $\boldsymbol{L}^1$ and $\boldsymbol{L}^2$ norm. The results are similar to the results observed for the boundary layer problem, and a gap of around a factor two exist between the optimal and the obtained convergences in $\boldsymbol{L}^1$-norm, whereas a nearly optimal convergence is obtained in $\boldsymbol{L}^2$ norm. We conclude that it is preferable to use the $\boldsymbol{L}^2$ norm when dealing with numerical solutions. For the rest of this thesis, we will only consider the $\boldsymbol{L}^2$ norm.



(a) L^1 error

(b) L^2 error

Figure 4.20: Comparison between $\boldsymbol{L}^1$ and $\boldsymbol{L}^2$ AMR for the numerical multiscale sinusoidal solution (equation (C.1)).

5.3 Potential flow around a circular cylinder

Finally we investigate the performance of the AMR method to accurately predict the steady solution for potential flow problems. In particular we consider the potential flow around a circular cylinder. The solution of this problem is well-known [131, 132]. A potential flow is characterized by its velocity potential ϕ deduced from the velocity $\boldsymbol{u}$:

$$\boldsymbol{u} = \nabla\phi \; . \tag{4.37}$$

For incompressible flows the velocity potential is solution of the Laplace equation

$$\Delta\phi = 0 \; . \tag{4.38}$$

The analytical solution is given in polar coordinates (r, θ) by

$$\phi = U_\infty \left(r + \frac{R^2}{r} \right) \cos(\theta) \; , \tag{4.39}$$

with U_∞ the freestream velocity norm, and R the cylinder radius.

The equation (4.38) is solved for a half-circle in a square domain of size $L_0 = 2$. The freestream velocity is set to $U_\infty = 1$ and the cylinder radius is $R = 0.05$. The cylinder boundary is modelized with an embedded boundary method [25].

An AMR study is performed with a security coefficient $\xi = 0.1$ and without minimal size constrain ($l_{max} = 15$). The error of the numerical solution is measured. The results are shown in figure 4.21, and examples of adapted meshes are shown in figure 4.22. The AMR method performs well: the adapted meshes provide an error close to the theoretical optimal error. However, without size constrain, the error follows strictly the optimal expectations, whereas the constrained results forms the already observed series of plateaus and is thus slightly less good. This is similar to the boundary layer problem results. Both these problems are single-scale examples. It is interesting to notice in figure 4.22 that the mesh size gradation from the cylinder to the far field is progressive, and that the two meshes are almost identical: the size constrain only increases the cell size in proximity of the boundary.

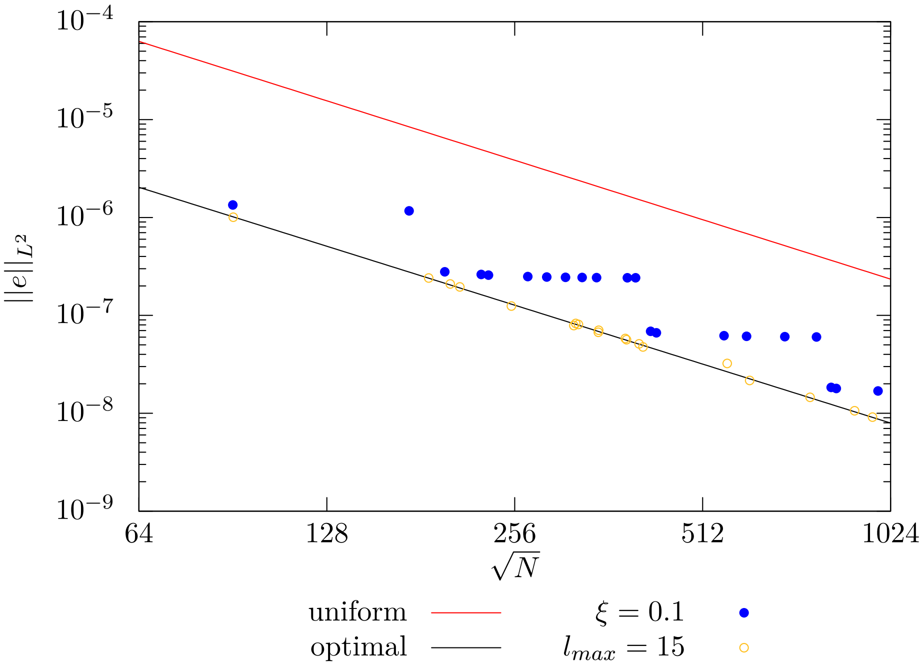
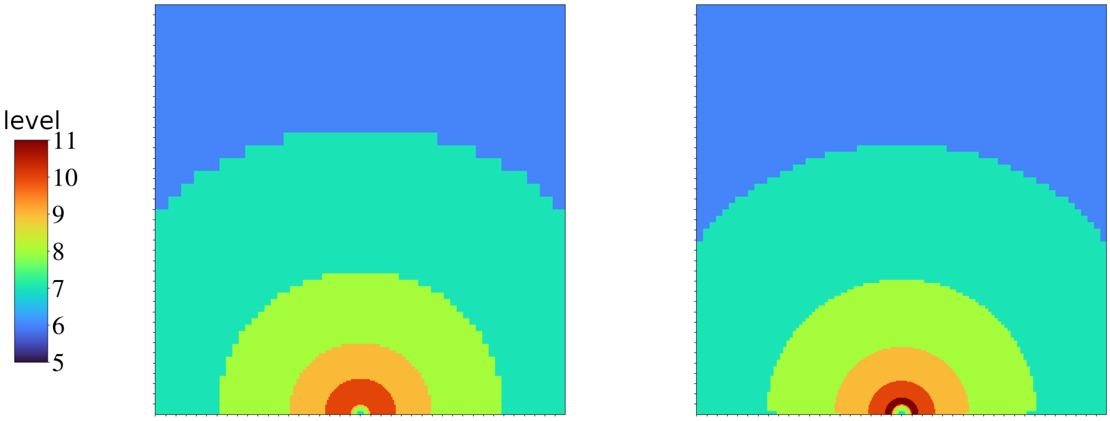


Figure 4.21: Error convergence of the current AMR method for the cylinder in a potential flow.



(a) Mesh levels for $l_{max}(\xi = 0.1)$. 39 116 elements.

(b) Mesh levels for $l_{max} = 15$. 40 708 elements

Figure 4.22: Meshes obtained for the cylinder in a potential flow.

6 Conclusion

In this chapter, we investigated first the behaviour of the numerical solution error when the second order Poisson-Helmholtz solver is used. Starting from an initial uniform mesh and coarsening the mesh, two main regimes of the error behavior may be observed. In the first regime, the numerical error is proportional to the interpolation error. This regime is composed of suboptimal grids in terms of interpolation error and both the interpolation and total error remain almost constant as the minimim element size is imposed and the number of elements varies. The existence of a second regime is associated to regions where the total error is almost equal to the theoretical minimal interpolation error. This regime is not general, but appears for some solutions that have multiple scales, and it is characterized by a dramatic increase in the total error when decreasing the number of elements. In that case, the numerical and the interpolation errors are decorrelated. To ensure an efficient mesh adaptation using local adaptation strategies, we must exclude the meshes entering into the second regime.

We have seen that we can define a characteristic mesh compression as the ratio between the number of elements of the mesh, and the number of elements that would contain a uniform mesh whose cell size is equal to the minimal cell size of the same mesh. The two identified error regimes are characterized by this ratio: if the compression ratio falls below a certain critical value that is problem dependent, the numerical and the interpolation errors are decorrelated, whereas they are proportional for compression ratio above this critical value. Based on these observations, a new error model is proposed for the Poisson-Helmholtz Basilisk solver, and an automatically computed minimal cell size prescription is explicitly imposed. It is based on an estimation of a limit compression ratio in order to exclude meshes in which the numerical and interpolation errors are decorrelated. The adaptive strategy is illustrated and validated on several test-cases.

5 | An AMR method for the solution of the incompressible Navier-Stokes equations

In the previous chapter, we proposed a new mesh adaptation method for the Poisson-Helmholtz equation. This method is based on a new error estimate used as refinement indicator and the introduction of an automatically computed minimal element size criterion to avoid overrefinement and bad quality meshes.

We propose in this chapter to extend this new approach to the case of the incompressible Navier-Stokes equations. First, we propose an error estimate for the Navier-Stokes equations as an extension of the previously defined AMR method for the Poisson-Helmholtz equation proposed in the previous chapter. Then, we apply it to several test-cases in order to validate the method and the hypothesis it relies on. The test-cases are representative for different flow field configurations including a Lamb-Oseen vortex, the flow around a circular cylinder at different Reynolds numbers, and a lid-driven cavity.

Contents

1 A simple error model for the Navier-Stokes equations 86
2 Application of AMR to different flows 86
  2.1 Lamb-Oseen vortex 87
  2.2 Flow around a circular cylinder 89
  2.3 Lid-driven cavity 104
3 Conclusion 106

1 A simple error model for the Navier-Stokes equations

Let us rewrite the Navier-Stokes system of equations (*cf.* chapter 2, section 2) for a steady incompressible viscous flow of kinematic viscosity ν, density ρ, velocity flow field $\boldsymbol{u}$, scalar pressure field p:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\,\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nabla \cdot (2\nu \boldsymbol{D}) \tag{5.1}$$

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p\right) = -\nabla \cdot ((\boldsymbol{u} \cdot \nabla)\,\boldsymbol{u}) \tag{5.2}$$

where the second equation is obtained by taking the divergence of the momentum equation and using the divergence free condition. Using the following identity [133]

$$(\boldsymbol{u} \cdot \nabla)\,\boldsymbol{u} = -\boldsymbol{u} \times \boldsymbol{\omega} + \frac{1}{2}\nabla \boldsymbol{u}^2 \ , \tag{5.3}$$

with $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$ the vorticity, we can rewrite equation (5.2) as the following Poisson equation

$$\nabla \cdot \nabla P = S \tag{5.4}$$

where $P = p + K$ is the dynamic pressure, with $K = \frac{1}{2}\rho \boldsymbol{u}^2$ the kinetic energy, and S represents the source term that can be expressed as

$$S = \rho \nabla \cdot (\boldsymbol{u} \times \boldsymbol{\omega}) = \rho \boldsymbol{\omega}^2 - \rho \boldsymbol{u} \cdot (\nabla \times \boldsymbol{\omega}) \ . \tag{5.5}$$

It is straightforward to see that $S = 0$ is zero for irrotational flows.

Equation (5.4) is interesting because P depends on both primary variables of the solver, pressure and velocity, and the equation remains valid for both, transient and steady flows. We propose to use equation (5.4) to estimate the error commited when solving numerically the incompressible Navier-Stokes system of equations. Note that this equation must be satisfied at the continuum level, but it is not directly used in the discretized numerical scheme (*cf.* chapter 2).

We use the error model for the Poisson-Helmholtz equation that we introduced in equation (4.12) in chapter 4. Writing the problem under matrix form and using the same notations for the matrix, we obtain the error

$$|P - P_h| \leq |\mathcal{D}^{-1}|\,|S - \Pi_h S| + |\mathcal{D}^{-1}\mathcal{P}|\,|P - \Pi_h P| \ . \tag{5.6}$$

Using the expression of P and the triangle inequality, we obtain the following error estimate

$$|P - P_h| \leq |\mathcal{D}^{-1}|\,|S - \Pi_h S| + |\mathcal{D}^{-1}\mathcal{P}|\,|p - \Pi_h p| + |\mathcal{D}^{-1}\mathcal{P}|\,|K - \Pi_h K| \ , \tag{5.7}$$

which combines interpolation errors on the pressure and on the kinetic energy. In the next sections, we investigate the efficiency of this error estimate used as refinement criterion in the mesh adaptation loop defined in the previous chapter for several test-cases.

2 Application of AMR to different flows

The efficiency of the novel AMR approach is investigated next for several flow problems. More precisely, we consider a Lamb-Oseen vortex solution of the Euler equations in section 2.1, several flows around a circular cylinder at different Reynolds numbers in section 2.2, and a lid-driven cavity problem in section 2.3.

2.1 Lamb-Oseen vortex

In this first test-case, we propose to study a high Reynolds number flow. When the viscosity is neglected, the steady incompressible Navier-Stokes equations reduce to the following steady incompressible Euler equations [132]:

$$\begin{cases} (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = -\frac{1}{\rho}\nabla p \\ \nabla \cdot \boldsymbol{u} = 0 \end{cases} \tag{5.8}$$

We propose to study a Lamb-Oseen vortex fixed at a given time t_0, which is a solution of the steady Euler equation. In polar coordinates (r, θ), it writes

$$u_\theta(r) = \frac{1 - exp(-(r/a_0)^2)}{r} \tag{5.9}$$

$$p(r) = \int_0^r \rho \frac{u_\theta(r)^2}{r} dr \tag{5.10}$$

and we impose the reference radius at t_0, $a_0 = 0.3$. The solution is shown in figure 5.1. The physical domain is $[-5, 5] \times [-5, 5]$. The analytical solution for pressure and velocity is imposed at boundaries. The simulation is initialized with the analytical solution, and the total error is measured on the numerical solution at time $t_{end} = 10$. The measured error combines the interpolation error and the natural, intrinsic, diffusivity of the numerical scheme.

The initial uniform mesh at $t = 0$ is transformed first using our AMR method with the analytical solution until the prescribed number of elements is reached. The initial solution is set on this initial adapted mesh. This procedure is followed because the maximal solution accuracy is fixed by the initial resolution.

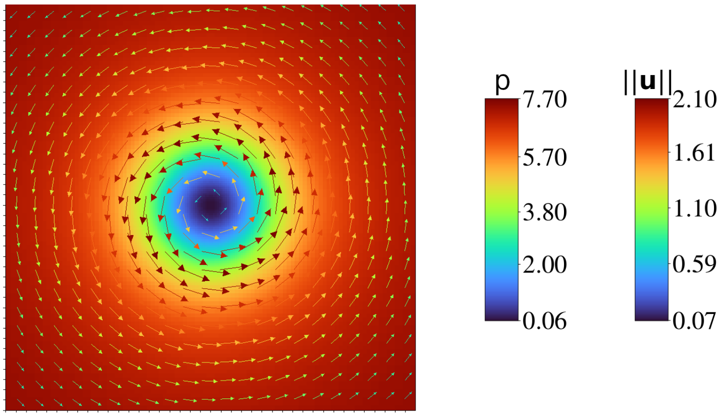
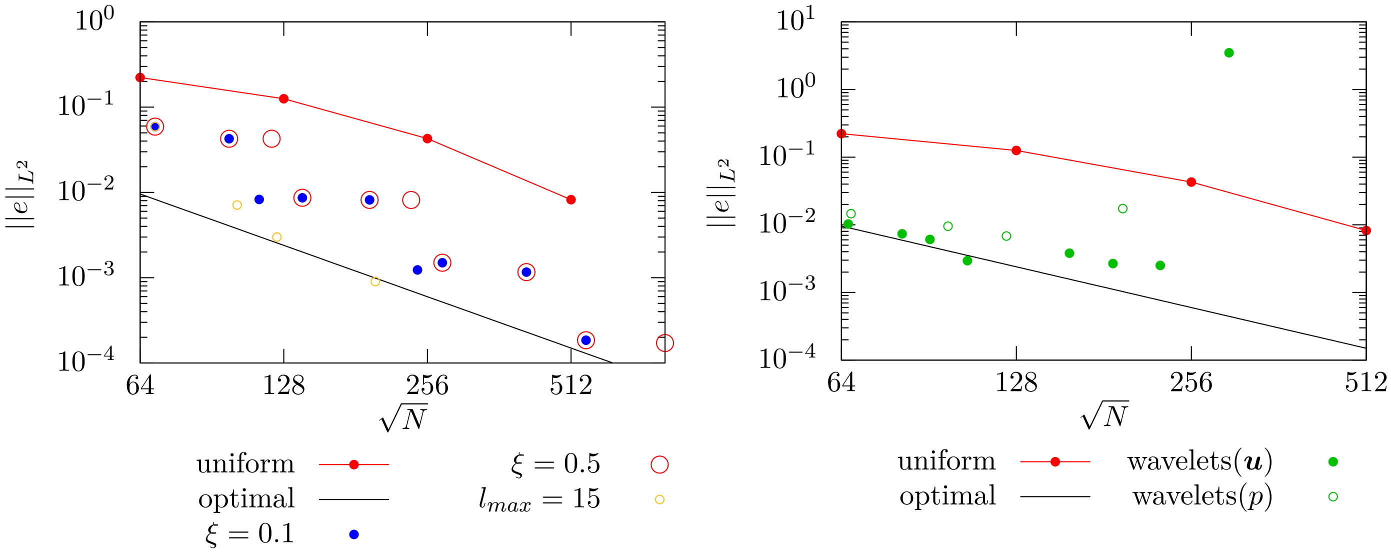


Figure 5.1: Zoom in $[-1, 1]^2$ of the solution: pressure field and velocity vector field.

Figure 5.2a compares the error on the pressure field obtained with our AMR algorithm for two security coefficients, $\xi = 0.1$ and $\xi = 0.5$, and without minimal cell size constrain ($l_{max} = 15$). The error on the velocity field is shown in appendix E, as it leads to the same conclusions than the error on the pressure field. The AMR performance are close to the optimal expectations for all three cases, especially for the case without minimal size constrain constrain. For the cases with the size constrain, the results are closer to the optimal with a security coefficient $\xi = 0.1$, than with $\xi = 0.5$, and the curve forms series of plateaus is obtained, which is similar to the observations made in the previous chapter and arises due to the discrete nature of the element size range. In figure 5.3 the meshes obtained for $N \approx 58\,000$

with the two security coefficient and without constrain are displayed. They show that the constrain and an increase in the value of the security coefficient limits the cell size range, which is expected regarding the results obtained in chapter 4. This test-case possesses only one scale (the scale of the vortex radius a_0). Thus, it confirms that the size constrain we introduce produces underperforming adapted meshes when dealing with non multiscale flows. Still, this underperformance is controlled and limited.

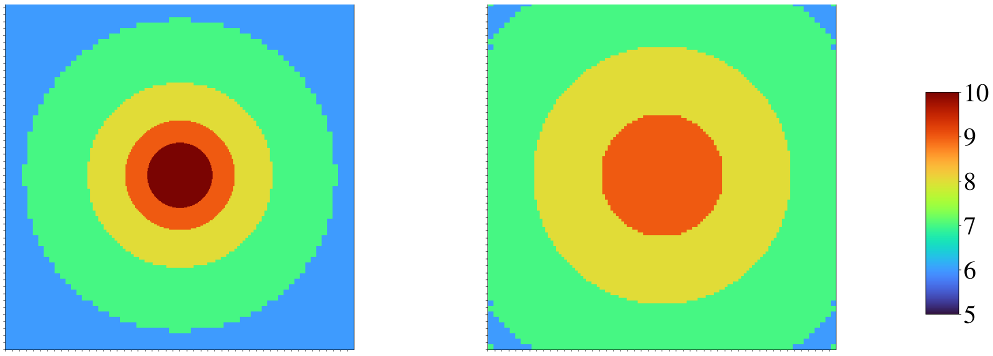
In figure 5.2b, we observe the results obtained with the wavelet-based AMR method of Basilisk when the velocity or the pressure field are used as an error criterion. The maximum level is set to $l_{max} = 14$, which is strictly larger than the maximum element level obtained in the resulting adapted meshes. The results are thus independent from the maximum level. It is remarkable to see that in both cases, the error is close to minimal interpolation error for a very low number of elements, but it remains almost constant or increases when the number of elements increases. Thus, the wavelet-based method is in that case not able to minimize the total error of the numerical solution when the minimal element size decreases. This shows the efficiency of the proposed AMR method in comparison with the wavelet-based method.



(a) comparison between different values of the security coefficient in the compression ratio-based AMR method.

(b) comparison between the wavelet-based AMR method using either p or $\boldsymbol{u}$ as an error criterion.

Figure 5.2: Error convergence on the pressure obtained for the Lamb-Oseen vortex solution of the Euler equations.



(a) $\xi = 0.1$, 59 560 element mesh

(b) $\xi = 0.5$, 56 032 element mesh

Figure 5.3: Adapted mesh levels obtained for the Lamb-Oseen vortex-like solution of the Euler equations in the domain $[-5, 5] \times [-5, 5]$.

2.2 Flow around a circular cylinder

Flows around a circular cylinder have been extensively studied over the last century. While simple, the circular geometry leads to complex flow structures which present very different behaviours depending on the experimental conditions. Moreover, this geometry is relevant up to a certain level of accuracy for numerous applications (Joukowski airfoil, offshore structures and pipelines, ...). Nowadays, many regimes have been identified for these flows [134, 135]: 6 regimes were identified by Achenbach [136, 137] based on experimental observations (*cf.* table 5.1), and they may be divided in 15 regimes based on boundary layer and wake behaviour considerations [138]. They can be categorized by their Reynolds number Re. We focus in this section on laminar flows fields ($Re \leq 150 - 300$). For $Re < 1$ the flow is dominated by diffusion phenomena. For $3 \lesssim Re \lesssim 40$, we observe a steady separation of the flow, and a steady recirculation zone in the wake of the cylinder. For $40 \lesssim Re \lesssim 300$, periodic laminar shedding occurs, and a Bernard von Karman vortex street is observed. In this last case, the flow is unsteady.

Flow regime	Re range
Creeping flow	$Re < 1$
Steady separation region	$3 - 5 < Re < 30 - 40$
Periodic laminar wake	$30 - 40 < Re < 150 - 300$
Subcritical regime	$150 - 300 < Re < 1.4\ 10^5$
Critical regime	$1.4\ 10^5 < Re < 10^6$
Supercritical regime	$10^6 < Re < 5\ 10^6$
Transcritical regime	$5\ 10^6 < Re$

Table 5.1: Regimes of flows around a circular cylinder [134, 135] categorized by their Reynolds number Re.

In the following, we consider a unit velocity flow ($u_\infty = 1$) around a circular cylinder at Reynolds numbers $Re = 0.05$, $Re = 0.5$, $Re = 40$, and $Re = 100$. For each case, the domain Ω lies in $[-64, 64] \times [-64, 64]$, and the cylinder of unit radius is centered at $(0, 0)$.

The boundary conditions are no-slip around the cylinder (dirichlet condition for the velocity $\boldsymbol{u} = (0,0)^T$ and neumann condition for the pressure $\frac{\partial p}{\partial n} = 0$ with $\boldsymbol{n}$ the normal to the boundary), a velocity inlet on the left boundary (dirichlet condition for the velocity $\boldsymbol{u} = (1,0)^T$ and neumann condition for the pressure $\frac{\partial p}{\partial n} = 0$), outflow boundary condition for the right boundary (null neumann condition for the velocity and null dirichlet condition for the pressure), and null neumann conditions elsewhere.

Except for the unsteady case $Re = 100$, the simulations are run on a uniform mesh of level $l = 8$ until the obtention of a steady solution. Then, the mesh adaptation begins at time t_{AMR} and the simulation runs until t_{end}, where a steady state solution is obtained on the adapted mesh. Preliminary tests suggested that the adaptation and final times must be set to $t_{AMR} = 150$ and $t_{end} = 300$. For $Re = 100$, the solution is unsteady, and the adaptation is started at the first iteration. The simulations runs until the periodic regime is obtained. For each case, the stability criterion is set to $CFL = 0.8$.

As no analytical solutions are available for the total velocity and pressure fields, we choose to focus on a global quantity and one quantity of interest to estimate the performance of the AMR method: the enstrophy $E = \int_\Omega \omega^2 dV$, and the drag coefficient $C_D = \frac{2}{\rho u_\infty^2 A}(F_v + F_p)$, with A the projected frontal area of the circle, $F_v = \int_S \mu \frac{\partial u_t}{\partial n} dS$ the viscous force on the surface S of the cylinder and $F_p = \int_S p\boldsymbol{n} \cdot \boldsymbol{dS}$ the pressure force. These quantities are not known exactly due to experimental and numerical uncertainties. For example Gautier *et al.* [139] report 17 different values taken from the literature for the drag coefficient of a cylinder at Reynolds $Re = 40$. These values belongs to the interval $[1.49, 1.62]$, with a mean value $C_D = 1.53$ and a standard deviation $\sigma = 0.04$. Because these values are not sufficiently precise to be used as a reference when performing error convergence studies, we estimate the error on these quantities as the difference between the measured quantity and a reference taken to be the mean value quantity measured on the finest meshes we obtained. We use the total enstrophy as a representative quantity of the whole system response, whereas the drag coefficient is only a local quantity integrated over the cylinder boundary. We note that the AMR method we propose reduces the error on the whole computational domain and it is not based on any output-based error criteria.

In the following sections we study the influence of the security coefficient ($\xi = 0.1$, $\xi = 0.8$, no security coefficient ($l_{max} = 15$)), and we compare the proposed AMR method with the wavelet-based AMR method ($l_{max} = 15$). In these results, we estimate the optimal error curve for the enstrophy obtaining numerically the $\boldsymbol{L}^1$ norm of the interpolation error for the squared vorticity ω^2 over the entire computational domain. It is estimated from the finest solution computed on uniformly refined meshes (maximum level $l_{max} = 12$). We can't estimate the optimal curve for the drag coefficient, as it is obtained from a surface integral: adjoint-based estimations are needed to treat this case.

2.2.1 Cylinder at Reynolds $Re = 0.05$

For a Reynolds number $Re = 0.05$, the flow is in the Stokes regime: viscous diffusion dominates the flow behaviour. In this regime, the streamlines follows the cylinder boundaries (*cf.* figure 5.4) and are almost symmetrical with respect to the axis $x = 0$. The pressure field presents a maximum localized in the front of the cylinder, and a minimum localized at its rear.

Figure 5.5, shows the enstrophy convergence results. From figures 5.5a and 5.5c, it may

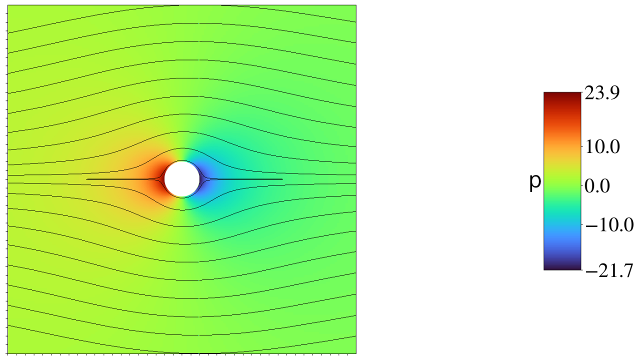
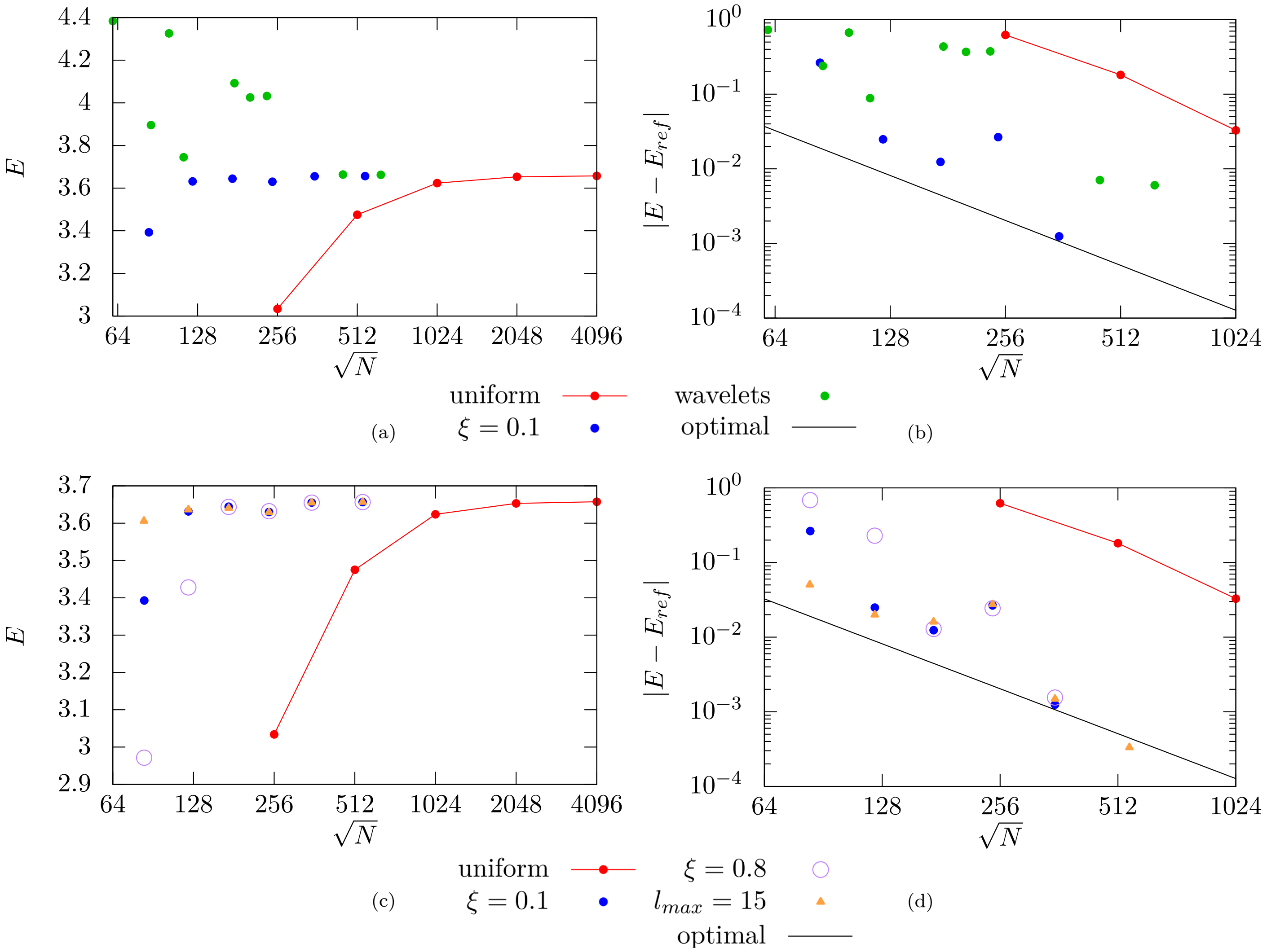


Figure 5.4: Pressure field and streamlines for a flow around a circular cylinder at Reynolds $Re = 0.05$. Zoom in the range $[-10, 10] \times [-10, 10]$.

be immediately observed that the enstrophy value is converged for uniformly refined meshes containing $N \approx 10^6$ elements, for wavelet-based AMR meshes containing $N \approx 262\,000$ elements and for the meshes obtained with our AMR method containing $N \approx 16\,000$. Thus, it is evident that for this problem, our AMR method is more efficient in comparison with uniform refinement and the wavelet-based method. This observation is confirmed by comparing the error convergence (figures 5.5b and 5.5d), where we can see that the error obtained with our method is closer to the optimal expectations than the wavelet-based AMR method.

The influence of the security coefficient in our method is visible in figure 5.5c: the lower the security coefficient is, the faster a converged value for the enstrophy is obtained: for $\xi = 0.8$, a nearly converged value is obtained for $N \approx 30\,000$ elements, which is further decreased for $\xi = 0.1$ ($N \approx 15\,000$) and when the compression ratio-based minimal cell size constrain is not used ($N \approx 7\,000$). It appears thus that for this case, the compression ratio constrain is not essential. An example of obtained meshes presented in figure 5.6 shows that an increase in the security coefficient leads to a reduced cell size range, and that the wavelet-based AMR method provides the biggest cell size range beyond the tested methods, which was expected.

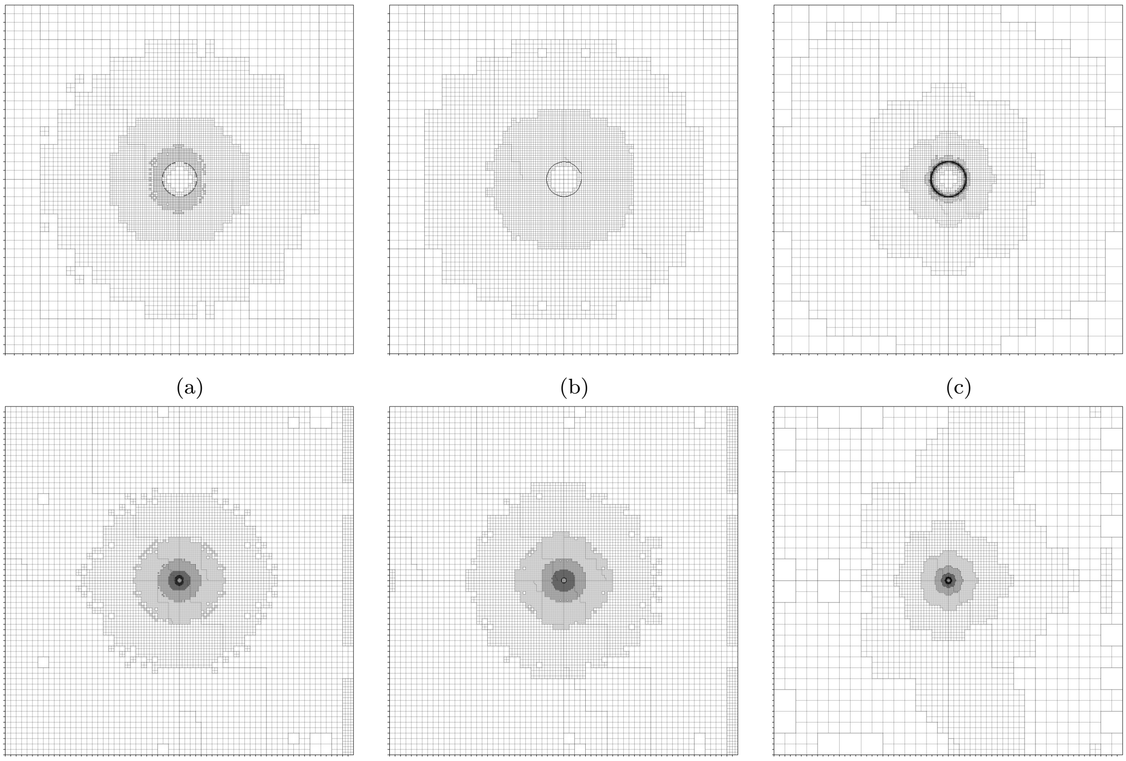
The results obtained for the drag coefficient are presented in figure 5.7. First, looking at the figure 5.7a, we see that the drag coefficient convergence properties with our AMR method is highly dependent on the the value of the security coefficient. In this particular case, the value of the drag coefficient does not seem to converge smoothly without compression ratio constrain ($l_{max} = 15$), whereas it seems to show much smoother behavior when adding the constrain and a security coefficient $\xi = 0.1$. With a security coefficient $\xi = 0.8$, the drag coefficient value seems to converge smoothly towards a constant value. Nonetheless, the compression ratio constrained AMR method seems to improve the accuracy of the drag coefficient using mesh adaptation. Finally, the values obtained on wavelet-based adapted meshes (figure 5.7b) have a wide dispersion for low number of elements, but begin to converge for higher number of elements. The difficulty to obtain accurate AMR results for the drag coefficient is not completely surprising: it is indeed known that adjoint-based error estimates are generally needed to best improve the accuracy of output functional with AMR methods [70, 71].



(a) (b)

(c) (d)

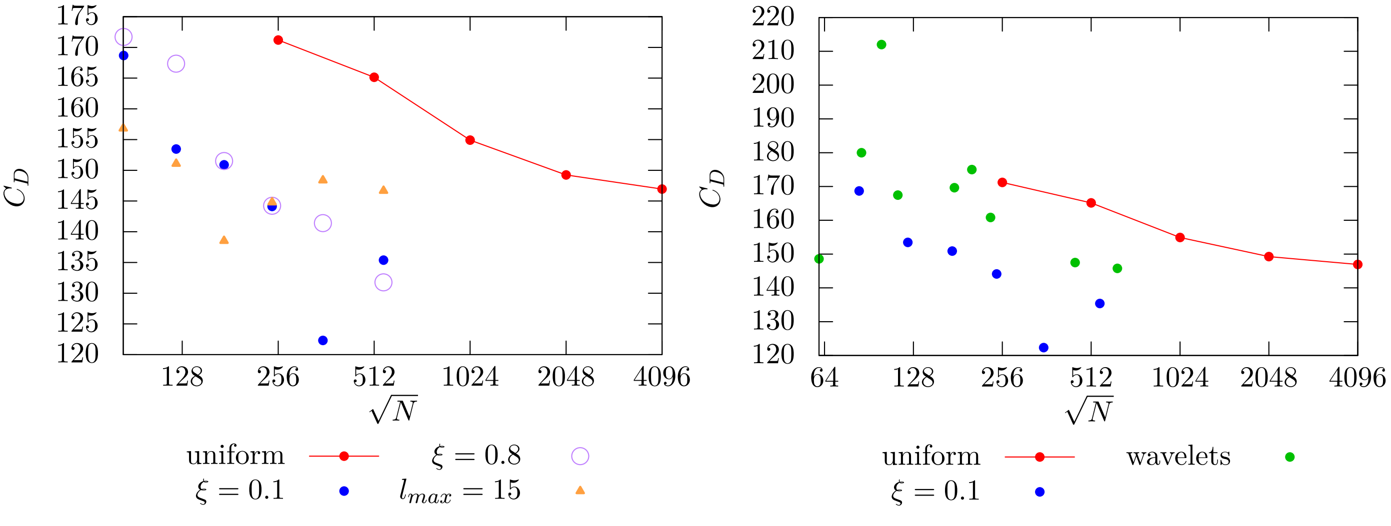
Figure 5.5: Enstrophy convergence for a flow around a circular cylinder at Reynolds $Re = 0.05$. Left: Enstrophy convergence. Right: Error convergence. Top: influence of the security coefficient. Bottom: comparison with the wavelet-based AMR method.



(a) (b) (c)

(d) (e) (f)

Figure 5.6: Meshes obtained for the flow around a circular cylinder at Reynolds $Re = 0.05$ with the proposed AMR method with $\xi = 0.1$ (left, $N = 15\,016$ elements), with $\xi = 0.8$ (middle, $N = 15\,004$ elements) and with the wavelet-based AMR method (right, $N = 15\,037$ elements). Top: zoom in the range $[-10, 10] \times [-10, 10]$. Bottom: complete domain $[-64, 64] \times [-64, 64]$.



(a) (b)

Figure 5.7: Drag coefficient convergence for a flow around a circular cylinder at Reynolds $Re = 0.05$. Left: influence of the security coefficient. Right: comparison with the wavelet-based AMR method.

2.2.2 Cylinder at Reynolds $Re = 0.5$

For a Reynolds number $Re = 0.5$, the flow around a circular cylinder is in an intermediate regime between diffusion dominated flow and an advection dominated flow. The streamlines still follow the cylinder boundaries, but they are no longer symmetric with respect to the axis $x = 0$. The region of maximum pressure in the front of the cylinder is reduced in comparison with the previous case, and the zone of minimum pressure in the wake of the cylinder is more widely spread, as illustrated in figure 5.8.

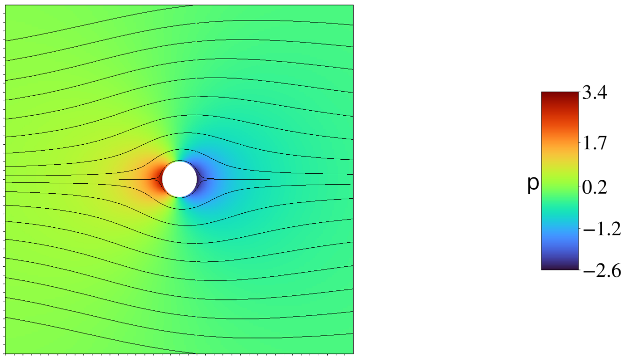
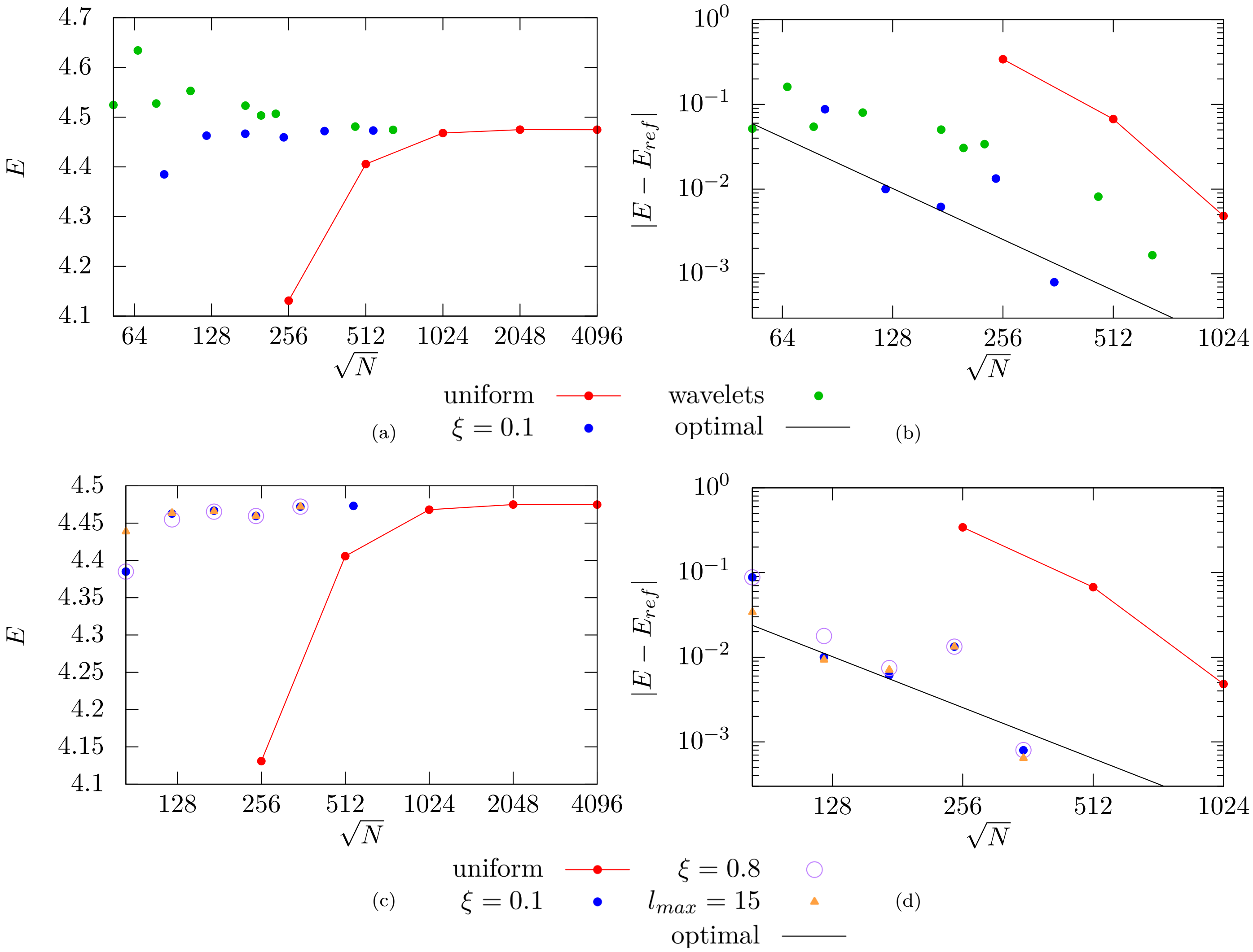


Figure 5.8: Pressure field and streamlines for a flow around a circular cylinder at Reynolds $Re = 0.5$. Zoom in the range $[-10, 10] \times [-10, 10]$.

In figures 5.9a and 5.9c, we observe the enstrophy convergence results. They are similar to the case $Re = 0.05$: $N \approx 10^6$ elements are necessary to obtain a converged value of the enstrophy using uniformly refined meshes, one order of magnitude less elements ($N \approx 10^5$) are necessary with the wavelet-based AMR method, and only $N \approx 16\,000$ elements are required with the compression ratio-based method. However, the influence of the compression ratio constrain is reduced in comparison with the previous case: the obtained results are almost identical with a security coefficient $\xi = 0.1$, $\xi = 0.8$ and without constrain ($l_{max} = 15$). These results are confirmed in the figures 5.9b and 5.9d, where we can see that the error obtained with our method is lower than the error obtained with the wavelet-based method, and that the error for different security coefficients is almost identical. Finally, the results obtained with our method are nearly optimal.

Unlike the $Re = 0.05$ case, the drag coefficient converges within the studied range of number of elements, as we can see in figures 5.10a and 5.10c. In particular, uniformly refined meshes provide an almost converged value of the drag coefficient for $N \approx 17 \cdot 10^6$ elements, whereas $N \approx 10^5$ elements are necessary for both the compression ratio-based and the wavelet-based AMR methods. Both AMR methods converge at order 2, as illustrated in figures 5.10b and 5.10d, but we can't specify which method is better than the other to minimize the error on the drag coefficient. The influence of the compression ratio constrain on the performance of the proposed method is small and concentrate observed only for meshes with a low number of elements, where less constrain leads to better results.

Examples of adapted meshes containing $N \approx 15\,000$ elements obtained with the compression ratio constrain for two security coefficients ($\xi = 0.1$ and $\xi = 0.8$) and with the wavelet-based method are shown in figure 5.11. It is first interesting to note that the meshes obtained with the two security coefficient are almost identical. The only significant difference is that a finer layer of small elements around the cylinder boundary is present when $\xi = 0.1$



(a) (b)

(c) (d)

Figure 5.9: Enstrophy convergence for a flow around a circular cylinder at Reynolds $Re = 0.5$. Left: Enstrophy convergence. Right: Error convergence. Bottom: influence of the security coefficient. Top: comparison with the wavelet-based AMR method.

and absent when $\xi = 0.8$. This does not impact the enstrophy value, which is identical between these two meshes ($E \approx 4.45$), and modifies only slightly the values of the drag coefficient ($C_{D,\xi=0.1} \approx 19.0$ and $C_{D,\xi=0.8} \approx 19.2$). The element size repartition is completely different for the wavelet-based adapted mesh. In comparison with our method, a much finer layer of fine elements is obtained around the cylinder, and bigger elements take place in the far field. It is interesting to note that this layer of very fine elements around the boundary is not sufficient to improve the estimation of the drag coefficient.

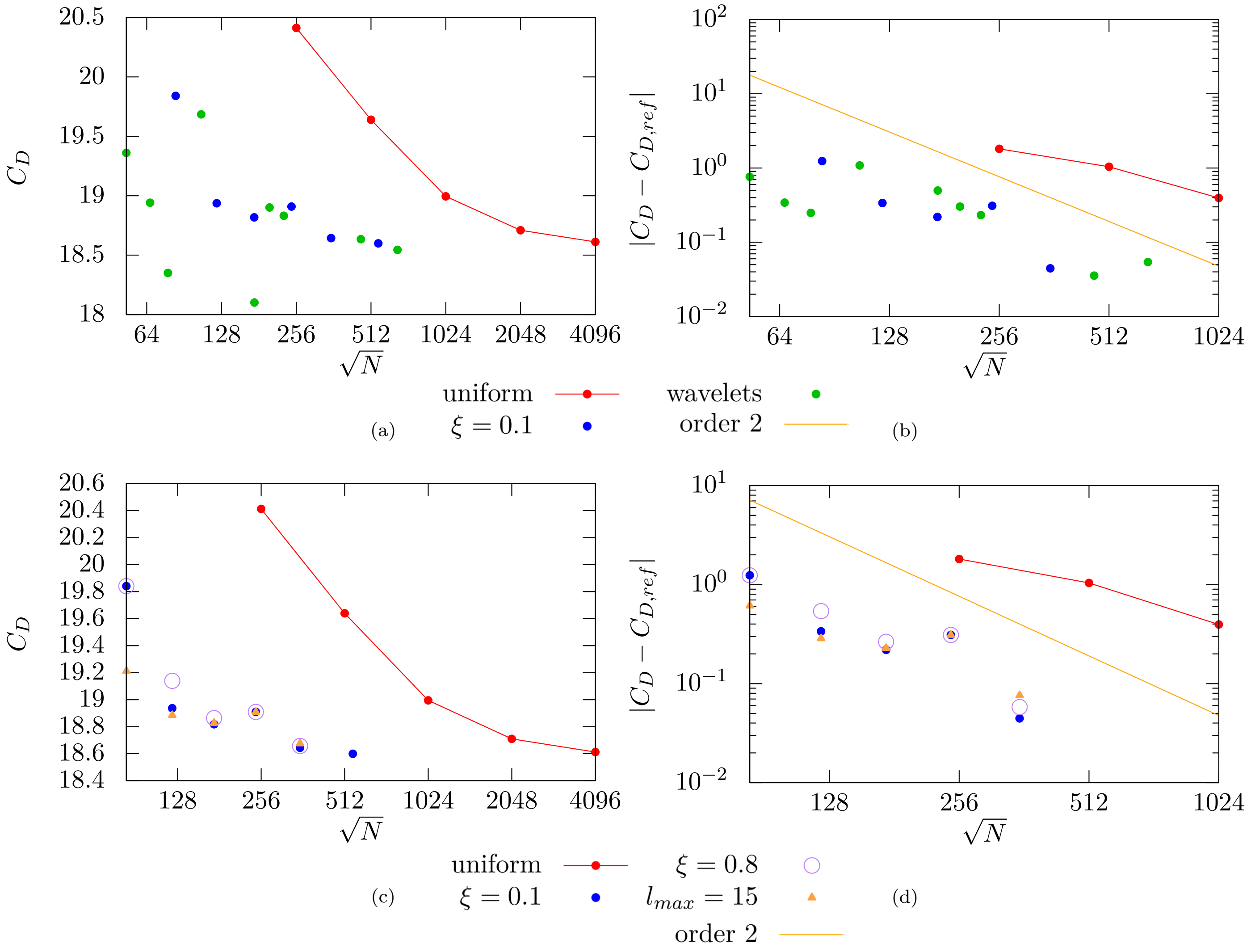


Figure 5.10: Drag coefficient convergence for a flow around a circular cylinder at Reynolds $Re = 0.5$. Left: Drag coefficient convergence. Right: Error convergence. Top: influence of the security coefficient. Bottom: comparison with the wavelet-based AMR method.

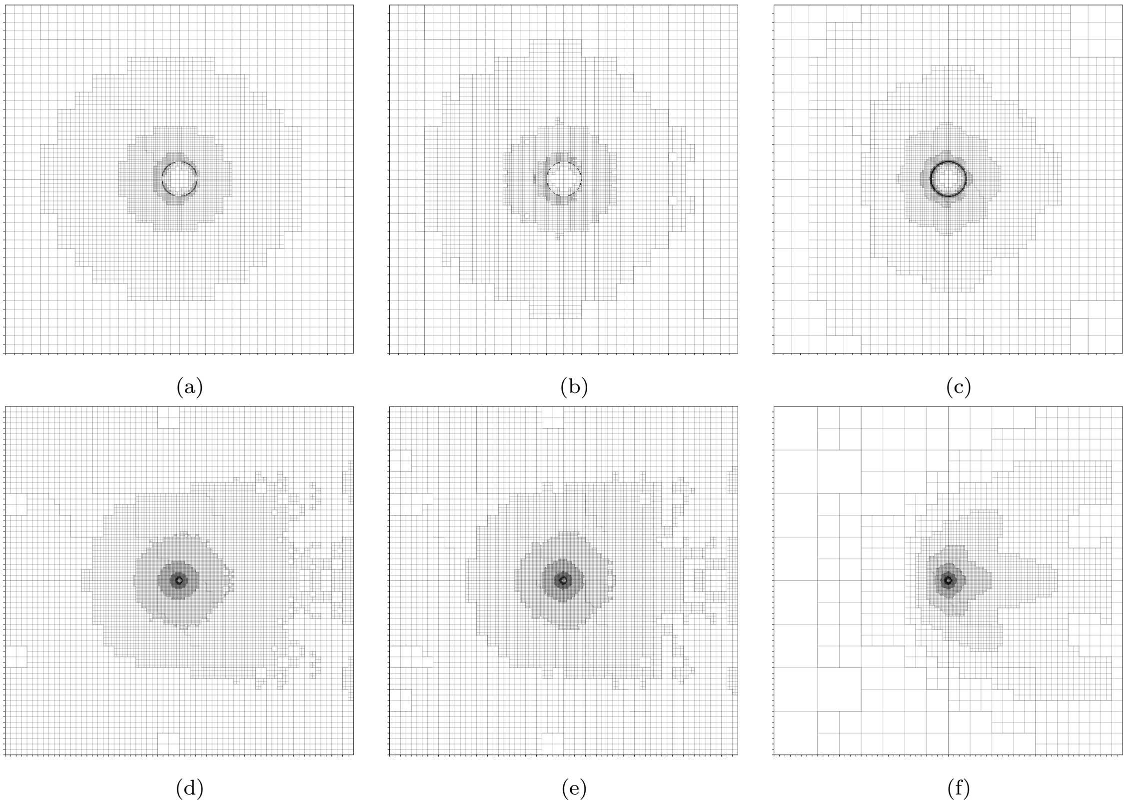


Figure 5.11: Meshes obtained for the flow around a circular cylinder at Reynolds $Re = 0.5$ with the proposed AMR method with $\xi = 0.1$ (left, $N = 14\,986$ elements), with $\xi = 0.8$ (middle, $N = 14\,992$ elements) and with the wavelet-based AMR method (right, $N = 11\,581$ elements). Top: zoom in the range $[-10, 10] \times [-10, 10]$. Bottom: complete domain $[-64, 64] \times [-64, 64]$.

2.2.3 Cylinder at Reynolds $Re = 40$

For a Reynolds number $Re = 40$, viscous effects are negligible everywhere except in the nearfield of the cylinder [132]. A steady recirculation zone exists in the wake of the cylinder (*cf.* figure 5.12), but the flow remains laminar. As in the previous cases, the maximum of pressure is located in front of the cylinder. However, two minimum of pressure are localized on the sides of the cylinder, which differs from the unique minimum present in the wake of the cylinder when $Re \leq 0.5$.

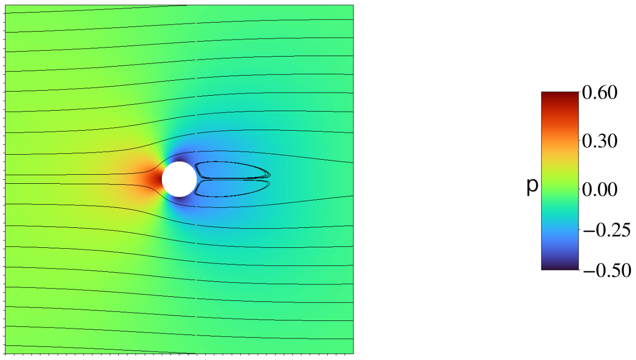
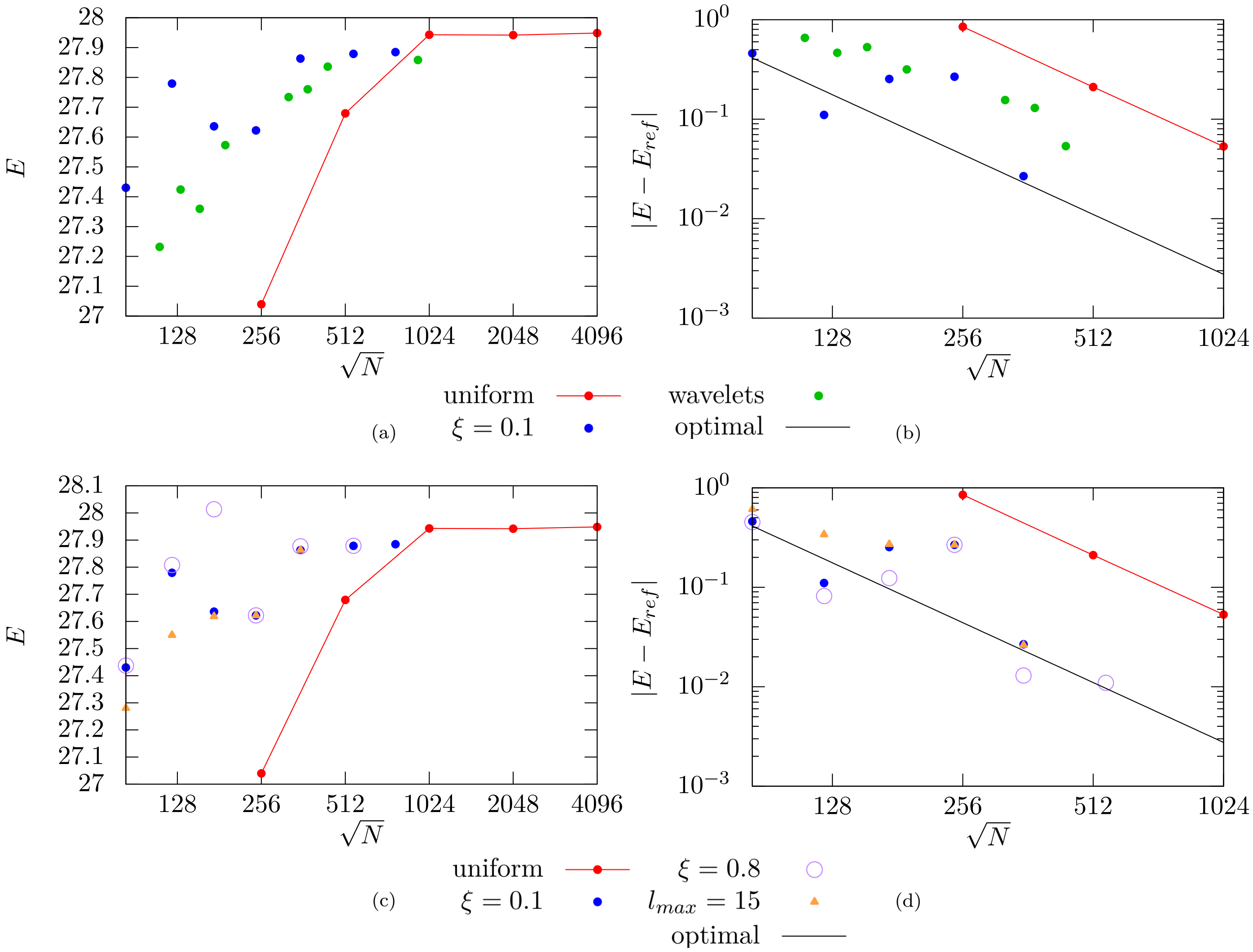


Figure 5.12: Pressure field and streamlines for a flow around a circular cylinder at Reynolds $Re = 40$. Zoom in the range $[-10, 10] \times [-10, 10]$.

Figure 5.13a shows that an almost converged value for the enstrophy is obtained for $N \approx 10^6$ elements on uniformly refined meshes, which is reduced to $N \approx 10^5$ elements for the wavelet-based and the compression ratio constrained methods. Nevertheless, the error obtained using the compression ratio based AMR method is lower than the error obtained with the wavelet-based method, as we can see in figure 5.13b. In figures 5.13c and 5.13d we can observe that for a low number of elements ($N < 20\,000$), the compression ratio constrain ($\xi = 0.1$ and $\xi = 0.8$) reduces the error in comparison with the unconstrained case ($l_{max} = 15$). For a higher number of elements, a small error reduction is observed when using a security coefficient of $\xi = 0.8$ compared to the two other cases, which give identical results. Thus, it seems that the additional constrain allows to reduce in a small extent the error of the numerical solution for the flow at Reynolds $Re = 40$.

The drag coefficient convergence is presented in figure 5.14. We may observe that the drag coefficient of all of the AMR methods tend to converge to a value ($C_D \approx 1.525$) higher than the converged value obtained on uniformly refined meshes ($C_D \approx 1.511$). Unfortunately the available results in literature are not known with a sufficient precision to asses which result is more precise [139]. One interpretation to this is that the results are not enough converged, and finer computations may be needed for AMR and uniform grids to converge. Another possibility is that for this particular case, the adaptation method used are all detrimental to the estimation of the drag coefficient. This is an existing possibility as the drag coefficient is a locally defined quantity, and the AMR methods we used are feature-based methods which try to reduce the error on the whole computational domain. It has been reported that feature-based error minimization process may lead to a convergence of local output-functional values toward a wrong solution [70, 71]. In such cases, the introduction of an adjoint-based AMR method is generally necessary.

Figure 5.15 compares meshes obtained with the compression ratio constrained method

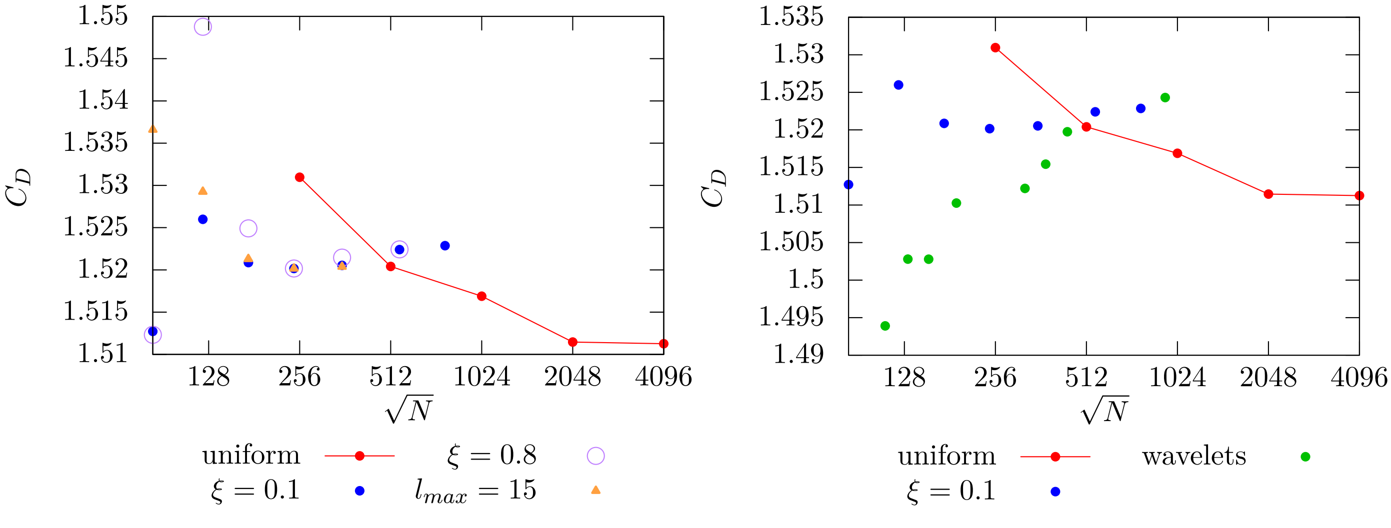


(a) (b)

(c) (d)

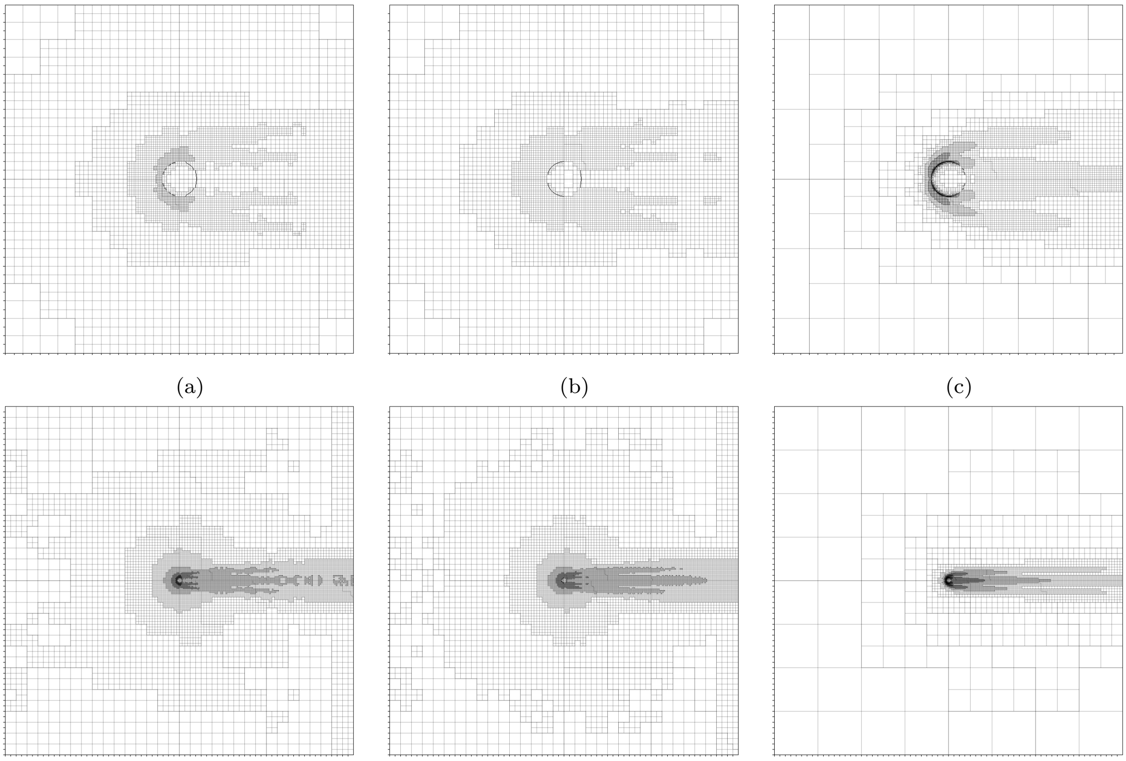
Figure 5.13: Enstrophy convergence for a flow around a circular cylinder at Reynolds $Re = 40$. Left: Enstrophy convergence. Right: Error convergence. Top: influence of the security coefficient. Bottom: comparison with the wavelet-based AMR method.

with two security coefficient ($\xi = 0.1$ and $\xi = 0.8$), and with the wavelet-based method. Despite the presence of a layer of finer elements around the cylinder when $\xi = 0.1$, which is absent when $\xi = 0.8$, the two meshes obtained with the compression ratio constrain show strong similarities. On the contrary, and as for the previous cases, the wavelet-based mesh is completely different in terms of element size repartition, and presents a very fast element size gradation in comparison with the new approach.



(a) (b)

Figure 5.14: Drag coefficient convergence for a flow around a circular cylinder at Reynolds $Re = 40$. Left: influence of the security coefficient. Right: comparison with the wavelet-based AMR method.



(a) (b) (c)

(d) (e) (f)

Figure 5.15: Meshes obtained for the flow around a circular cylinder at Reynolds $Re = 40$ with the proposed AMR method with $\xi = 0.1$ (left, $N = 14\,992$ elements), with $\xi = 0.8$ (middle, $N = 14\,995$ elements) and with the wavelet-based AMR method (right, $N = 12\,250$ elements). Top: zoom in the range $[-10, 10] \times [-10, 10]$. Bottom: complete domain $[-64, 64] \times [-64, 64]$.

2.2.4 Cylinder at Reynolds $Re = 100$

For a Reynolds number $Re = 100$, the flow around a circular cylinder is unsteady and a periodic formation of vortices (Karman vortex street) is observed in the wake the cylinder [132, 140], such as illustrated in figure 5.16. This is a preliminary test-case which shows a first application of our AMR method to unsteady flows. We perform six AMR simulations using our compression ratio constrained method with a security coefficient $\xi = 0.1$. The number of elements obtained at the end of each simulation is shown in table 5.2. As a comparison, we also run four simulations on uniformly refined meshes (cell level $l \in [\![9, 12]\!]$).

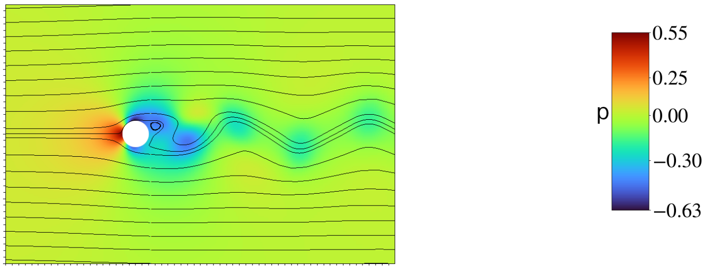


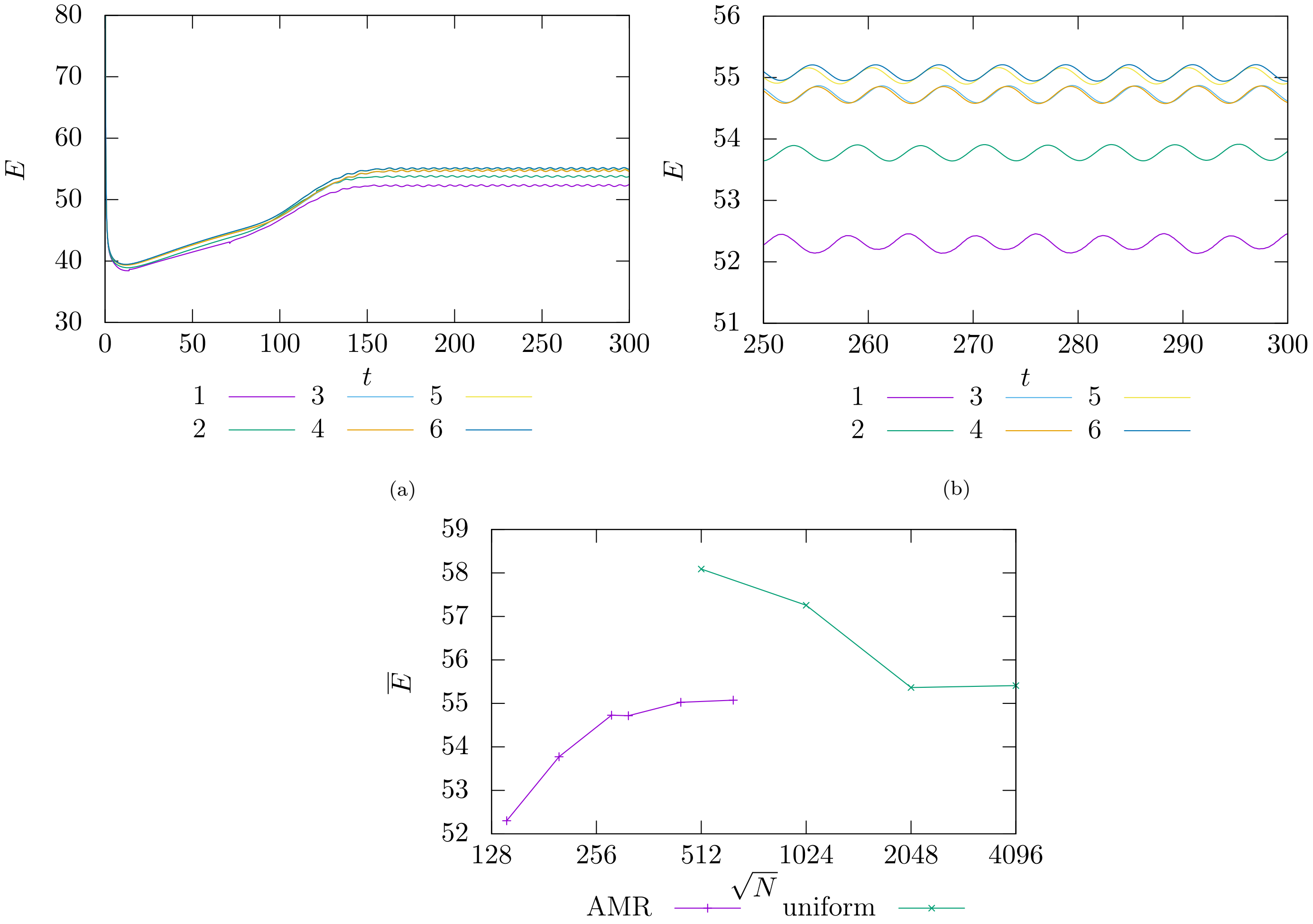
Figure 5.16: Pressure field and streamlines for a flow around a circular cylinder at Reynolds $Re = 100$ at time $t_{end} = 300$. Zoom in the range $[-10, 20] \times [-10, 10]$.

Figure 5.17 shows the enstrophy results obtained for the AMR simulations. It may be observed that a periodic state with a steady mean value is obtained at the end of the simulation. As it could be expected, the value of the enstrophy oscillate at a frequency $f \approx 0.16\,\text{Hz}$ corresponding to twice the vortex shedding frequency estimated from [132, 140]. In figure 5.17c, the mean enstrophy obtained on adapted meshes is compared with the mean enstrophy value obtained on the uniformly refined meshes. One can observe that this value converges as the number of element increases to the same value on both, AMR and uniform meshes. However, $N \approx 2 \cdot 10^5$ elements are sufficient when using mesh adaptation, and $4 \cdot 10^6$ elements are needed on uniformly refined meshes, which shows the efficiency of the adaptation method in comparison with uniform meshes.

The drag coefficient evolution in time and convergence in number of elements is shown in figure 5.18. At the end of the simulation, the drag coefficient follows a periodic evolution in time around a constant mean value at twice the vortex shedding frequency. The value of the drag coefficient is almost constant on adapted constant in function of the number of elements: it varies between $C_D = 1.332$ and $C_D = 1.334$ when $N \geq 40\,000$ elements. As we can see in figure 5.18c a similar value for the drag coefficient is obtained for uniformly refined meshes containing at least $4\,000\,000$ elements. Thus, the drag coefficient seems correctly captured with very few elements using mesh adaptation than on uniformly refined meshes. Combining these observations with the observations on the enstrophy, we observe that the proposed AMR method is efficient for this unsteady problem, even if it does not integrate the time discretization errors.

case	1	2	3	4	5	6
N	20 005	40 000	80 017	100 019	200 026	400 030

Table 5.2: Number of elements obtained at the end of each AMR simulation for a circular cylinder at Reynolds $Re = 100$.



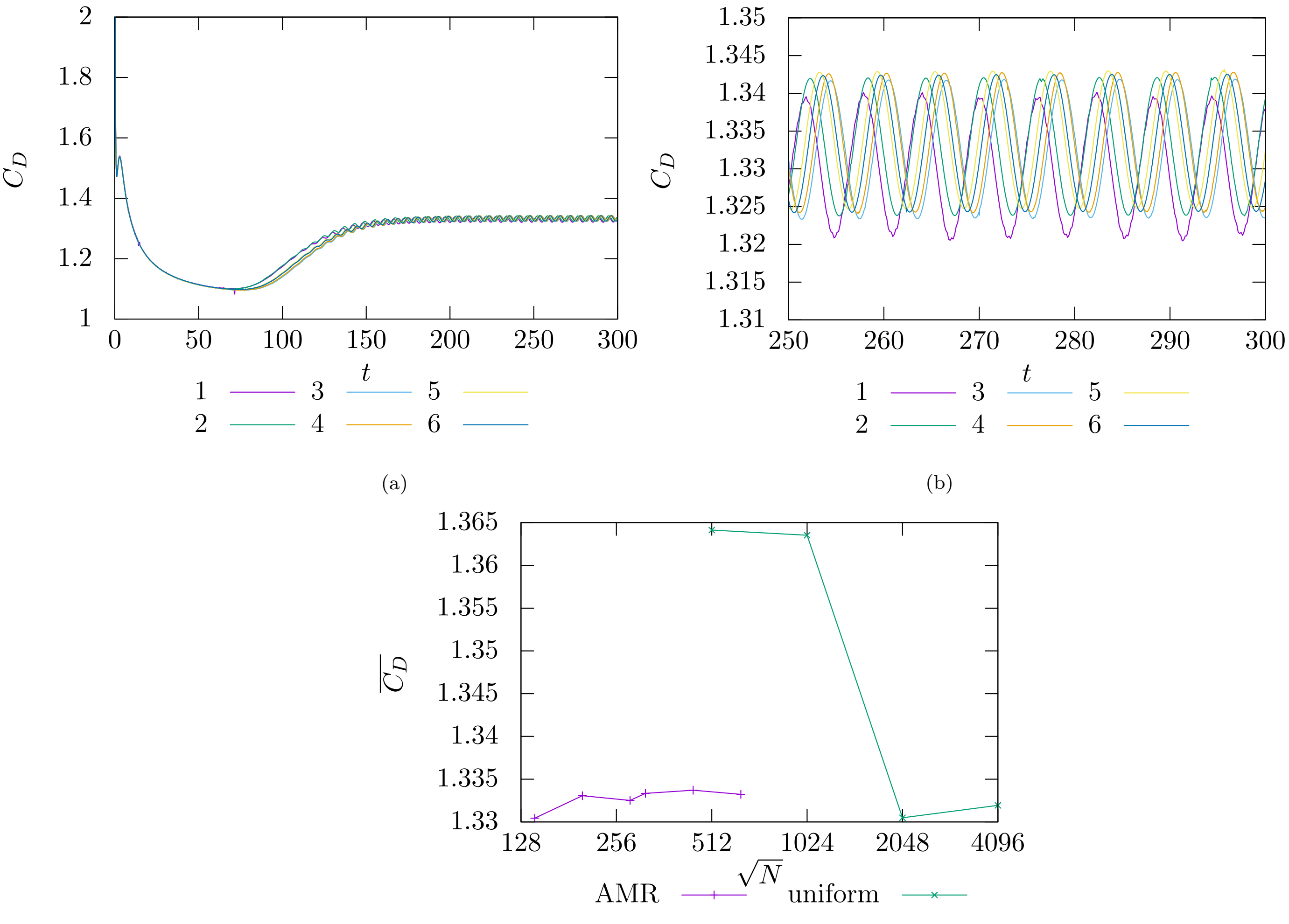
(a)

(b)

(c)

Figure 5.17: Top: enstrophy evolution in time. Bottom: mean enstrophy convergence.

Finally, we observe in figure 5.19 the obtained adapted mesh containing around 200 000 elements. It must be noticed that the evolution of this adapted mesh is steady at the end of the simulation, even if the simulation is unsteady. In particular, once the steady state is reached, the adapted mesh does not evolves in time, on the opposite to the periodic vortices and the streamlines, and it does not follows each vortex taken individually. Adding to that the obtained element size repartition in the wake of the cylinder, it seems rather logical to think that the mesh is adapted for the mean flow field, and that it is not necessary to accurately describe individually each periodic vortex to obtain an efficient adapted mesh.

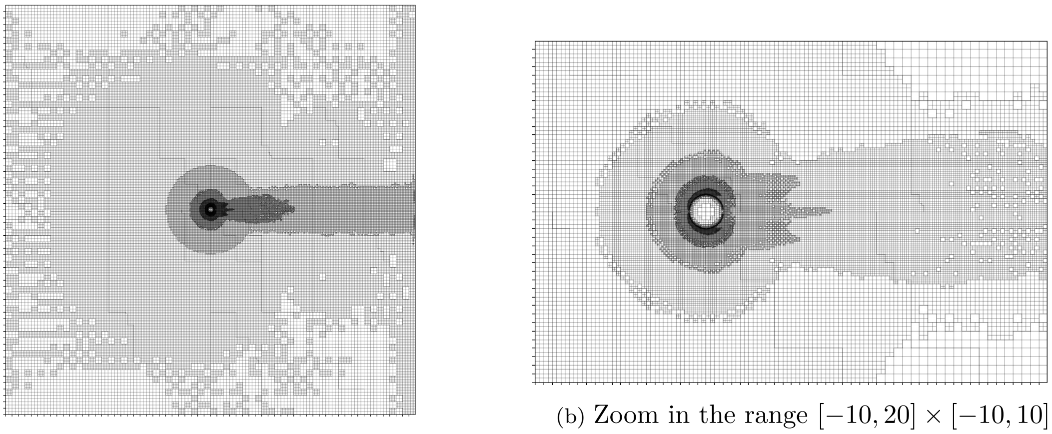


(a)

(b)

(c)

Figure 5.18: Top: drag coefficient evolution in time. Bottom: mean drag coefficient convergence.



(a) Full domain $[-64, 64] \times [-64, 64]$.

(b) Zoom in the range $[-10, 20] \times [-10, 10]$.

Figure 5.19: 200 000 elements mesh for the flow around a circular cylinder at Reynolds $Re = 100$.

2.3 Lid-driven cavity

In this section, we apply our AMR method on the well-known lid-driven cavity test-case [141, 142, 143, 144], which has been used numerous times in order to assess the performance of numerical schemes. This problem studies the flow generated by a moving bounding wall inside of a closed rectangular domain. While this case has simple geometry and boundary conditions, the obtained flow field presents some difficulties, and in particular singularities in the velocity field in proximity of the two corners of the moving boundary, due to the imposed boundary conditions (unit tangential velocity at the moving boundary *vs* null velocity at the side walls). The consistency of the results obtained by numerous studies and the analogy with analogous problems show that the effect of these singularities should be only a local around the corresponding corners, and have a negligible impact on the solution in the rest of the domain [143, 144]. In the present case, we consider a flow at a Reynolds number $Re = 1000$, which reaches a steady state presenting a major central vortex and secondary vortices in the corners, such as represented in figure 5.20. After a short description of the numerical setup, we compare two velocity profiles obtained when the steady state is reached with the numerical results of Ghia *et al.* [141].

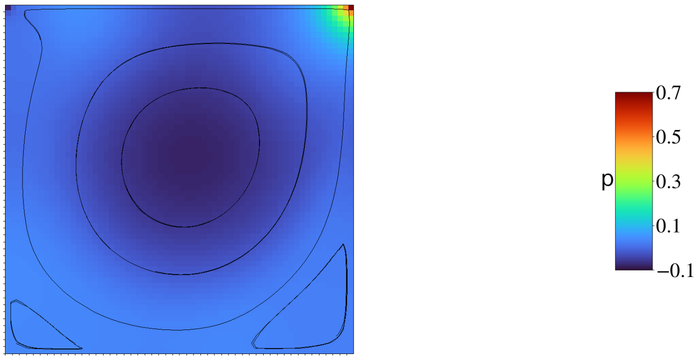
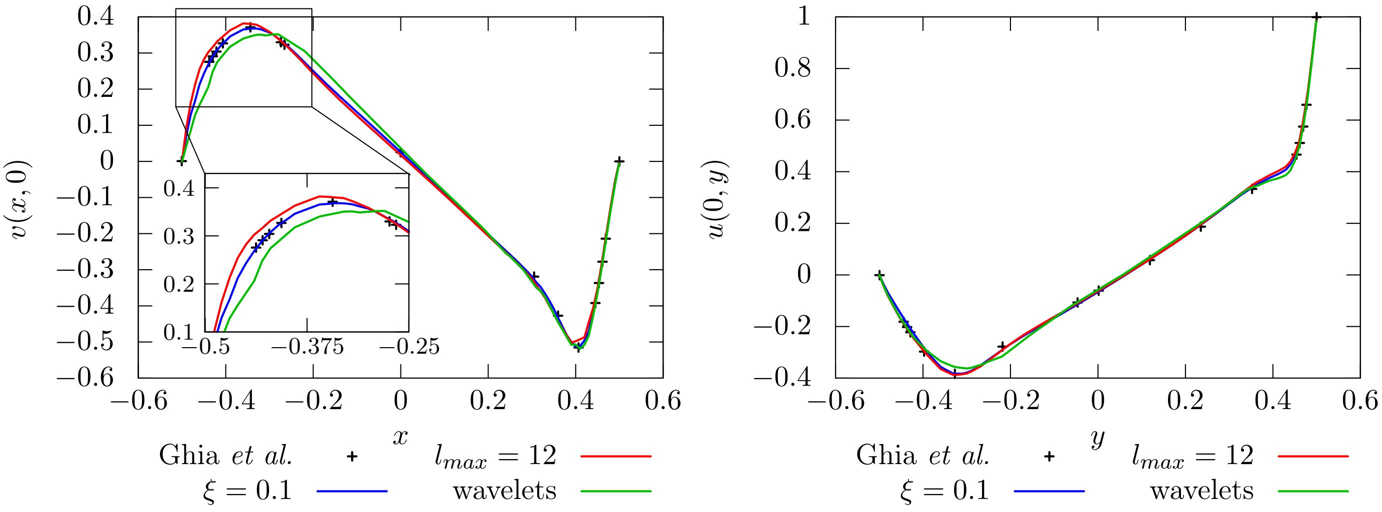


Figure 5.20: Pressure field and streamlines for the lid-driven cavity at Reynolds $Re = 1\,000$.

The cavity is a square of unit length positioned in the intervals $[-0.5, 0.5] \times [-0.5, 0.5]$. The simulation is run until a steady state is reached ($t_{end} = 44$). The left, right and bottom boundaries are no-slip boundaries, and the top boundary imposes a unit horizontal velocity. The compression ratio-based AMR method is parametrized with the objective number of element $N = 2\,800$ and the security coefficient $\xi = 0.1$. The same algorithm is tested without constrain on the minimal cell size (for practical reasons, the maximum level is set to $l_{max} = 12$, which is strictly greater than the effective maximum level obtained on the resulting mesh). The adaptation process is performed at each iteration. We compare the results obtained with our AMR approach with the ones obtained using the wavelet-based method whose sensor is the velocity field and the maximum level set to $l_{max} = 15$.

Defining the velocity $\boldsymbol{u} = (u, v)^T$, the interpolated velocity profiles at the center-lines $u(0, y)$ and $v(x, 0)$ are shown in figure 5.21 in comparison with the reference data [141]. It is clear that the three mesh adaptations perform differently: the compression ratio-based method ($l_{max}(\eta_{\xi=0.1})$) provide an accurate solution that matches very good the reference solution, whereas the solutions obtained with the wavelet-based method and the unconstrained method show a misalignement of the results with the reference – especially for the velocity profile $v(x, 0)$ for $x \leq -0.2$. Thus, two conclusions may be drawn: our interpolation error based AMR method shows better performance than the wavelet-based AMR method, and the minimal cell size constrain has a significant impact in the case of the lid-driven cavity.

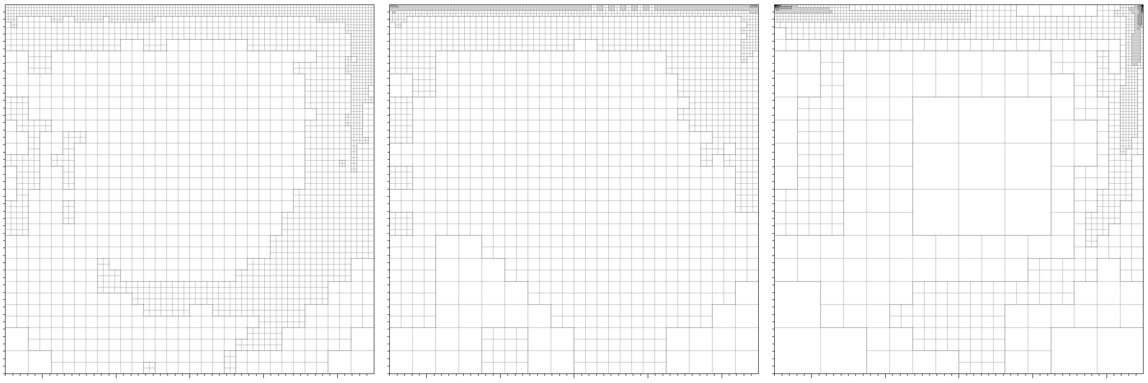


(a) $v(x,0)$ profile.

(b) $u(0,y)$ profile.

Figure 5.21: Comparison between the velocity profiles $\boldsymbol{u} = (u,v)^T$ obtained for three $N \approx 2\,800$ elements adapted meshes and the reference. The compression ratio-based constrained methods ($l_{max}(\eta_{\xi=0.1})$) matches the reference profiles. The same AMR procedure without constrain ($l_{max} = 12$) and the wavelet-based methods are less accurate.

The resulting adapted meshes are presented in figure 5.22. The mesh obtained with the compression ratio-based method ($l_{max}(\eta_{\xi=0.1})$) contains $N = 2\,794$ elements, the unconstrained mesh contains $N = 2\,800$ elements and the wavelet-based adapted mesh contains $N = 2\,839$ elements. They are all very different. The compression ratio-based adapted mesh present a layer of cell with a uniform size (level $l = 7$) along the top boundary, and a increasing element size in the rest of the domain. The mesh is composed with only four different element size, and only six elements reach the maximum cell size ($l = 4$). The unconstrained mesh has finer cells in the two top corners ($l = 9$) than in the rest of the top boundary ($l = 8$). The element size increases in the rest of the domain, and more elements with the biggest size ($l = 4$) are obtained than in the constrained mesh. The wavelet-based adapted mesh presents two points (the two top corners) where all of the finest elements (up to level $l = 15$) are concentrated, and the rest of the domain is filled with much bigger elements (with a minimum level $l = 3$). On the opposite to the two other cases, the top boundary is represented with a large variety of element size. These behaviours are interesting: the corner singularities and the main vortex correspond to the two main scales of the flow field, and they are different in terms of both the amplitude and the spatial extension. This is thus a multiscale flow. The behaviour of the adapted mesh is similar to the multiscale sinus (appendix C): the best mesh ($l_{max}(\eta_{\xi=0.1})$) limits the amplitude of the cell size range and resolves all the scales of the flow accurately. On the opposite, the less efficient meshes (unconstrained AMR and wavelet-based AMR methods) are over-resolved in the region of the scale with the main amplitude (the two top corners) and under-resolved for the rest of the flow. Thus, the compression ratio-based constraint seems particularly relevant for multiscale flows.



(a) $l_{max}(\eta_{\xi=0.1})$, $N = 2\,794$ elements. (b) $l_{max} = 12$, $N = 2\,800$ elements. (c) wavelet-based AMR, $N = 2\,839$ elements.

Figure 5.22: Comparison between the meshes obtained with the compression ratio-based and the wavelet-based methods for a lid-driven cavity at Reynolds $Re = 1\,000$.

3 Conclusion

In this chapter, we developed a new error estimate for the steady incompressible Navier-Stokes equations. It is based on a form of the Navier-Stokes equations obtained by taking the divergence of the momentum equation, applying the divergence free condition and rewriting the advection term introducing the vorticity and the kinetic energy. These operations result in a Poisson equation for the dynamic pressure. After introducing the error model developed for a Poisson-Helmholtz equation in chapter 4, we obtain an error model based on the interpolation error of the pressure and the kinetic energy for the Navier-Stokes equations.

This error estimate is used with the compression ratio-based minimal element size constrain developed in chapter 4 and tested on several test-cases: a Lamb-Oseen vortex, a circular cylinder at various Reynolds numbers and a lid-driven cavity. They show three points: the proposed method is nearly optimal for incompressible Navier-Stokes solutions, it outperforms the wavelet-based algorithm in terms of error minimization, and the minimal element size constrain is mainly relevant for multiscale flows.

6 | Conclusion and perspectives

This thesis is placed in the context of a continuous effort to improve the performances of the CFD free-software Basilisk, and is a first step toward the long-term objective of the introduction of an efficient AMR method capable of dealing with compressible multiphase flows. A new AMR strategy able to reduce the total error of the numerical solutions of the Poisson-Helmholtz and incompressible Navier-Stokes equations is developed and implemented. The proposed AMR method is based on the Riemannian metric theory extended to quad/octree grids where an additional constrain for the minimum element size is considered as well as an associated error estimation.. The introduction of this additional constrain in the minimization problem is motivated by the observation that for some problems there exists a family of grids in which the error introduced in the numerical solution does not show correlation with the local interpolation error. The total and the interpolation errors are correlated only for meshes where the compression ratio is above a problem dependent critical value. The additional constrain on the minimum element size is shown to effectively restrict the search domain to grids where the numerical error is more likely to be proportional to the interpolation error. This minimum element size is computed theoretically and shown to be a function of the Hessian of the solution and the number of grid points imposed and free of any other user-defined parameter. In practice, a user-defined security coefficient is introduced to control the performance of the algorithm. Finally, a model is proposed to extrapolate the method proposed to the solution of the incompressible Navier-Stokes equations.

Chapter 2 describes the main components of the software Basilisk and illustrates some limitations of the wavelet-based AMR method which motivated the current work and the need to exploit the full potential of the Basilisk mesh adaptation capabilities.

The continuous mesh framework, a continuous equivalent to discrete meshes, is introduced in chapter 3. It provides the estimation of the minimal interpolation error on simplicial elements through the resolution of an optimization problem, and we extend the development to square/cubic elements. This theory allows us to obtain a theoretical estimation of the minimal interpolation error on Basilisk quad/octree meshes without any undetermined constant, and to use it as a reference to evaluate the performance of mesh adaptation methods.

In chapter 4, we use numerical experiments to explore the range of applicability of adaptation strategies based on a local measurement of the interpolation error and to propose a new AMR method. Using the compression ratio, defined as the ratio between the minimal cell size and the mean cell size of a mesh, we show that an optimal value for this ratio exists which minimizes the total error of the solution. Below this critical value, the numerical errors are no longer correlated to the interpolation errors and local AMR strategies are not effective to reduce the numerical error. Thus, the optimal mesh in terms of pure interpolation error does not necessarily minimize the error of the numerical solution, and it is possible to identify examples where the performance of this mesh can be detrimental in comparison with the solution obtained on uniformly refined meshes. In an attempt to exclude grids where the total numerical error is not correlated with the interpolation error, we propose two different algorithms to limit the mesh compression through an automatic estimation of the minimal cell size which is explicitly imposed during the AMR procedure. The first algorithm is based on the tree structure, while the second method proposed is based on the Riemannian metric theory. This later method shows that the additional constraint is an intrinsic property of the analytical function. This replaces the traditional user-imposed minimal element size imposed in Basilisk, and acts as an additional constrain in the error minimization problem: the obtained adapted mesh is no longer an optimal mesh toward the interpolation error, but a sub-optimal mesh in terms of the interpolation error while leading to a dramatic decrease of the total error in some problems.

Finally, we propose two error estimations based on the discretized form of the Poisson-Helmholtz (chapter 4) and the incompressible Navier-Stokes equations (chapter 5). For the Poisson-Helmholtz equation, the final error estimation is a mean of interpolation errors in a local stencil. The importance of using at least a $\boldsymbol{L}^2$ norm error estimation when dealing with numerical solutions is observed and quantified. Then we propose a Navier-Stokes error model as an extension to the Poisson-Helmholtz error model in order to minimize the error on both the pressure and the kinetic energy. Intrinsically, this error model does not count for time discretization errors when solving the unsteady Navier-Stokes equations, but its performance is assessed for both steady and unsteady problems with satisfactory results.

Perspectives

The works done in this thesis provide a series of tools toward automatic mesh adaptation for complex flows solved by the Basilisk solver. With this objective in mind, numerous extensions may be investigated, which may finally end with numerical investigation of unsteady multiphase compressible flows, such as the ones observed in the CACHMAP project [7, 8].

First, the method proposed in this PhD to provide the additional constrain on the minimum grid size on some problems remains heuristic. Indeed we have reported problems where the introduction of the additional constrain penalizes the performance of the method. The reasons behind why we need to add this constrain and the circumstances where this constrain need to be added have not been clarified. In order to advance on the characterization of the problem reported, it would be interesting to check if the issues reported in this manuscript can be reproduced in other solvers based on alternative resolution methods. Another idea would be to check the relation between the total error, the interpolation error and the residual of the equation on the various embed grids defined in the tree-based algorithm. These numerical experiments may provide clues about how to predict when the correlation between the interpolation and the total errors contained in the solution is lost and may provide a better criterion to reduce the total numerical error.

For fully unsteady flows, the possibility to use the proposed error estimate developed for the Navier-Stokes equations must be further investigated, especially for complex unsteady problems such as flows with turbulent behaviour, separation, etc. From a pure mathematical point of view, the integration over a given period of time necessitate the introduction of the time dependency of the error estimate. Thus, it would be interesting to enrich the proposed error estimate with time-errors contribution or at least investigate the behavior over time of our error estimate when AMR is applied at different time frequencies.

Multiphase flows are represented in Basilisk through the VOF method [111]: it uses a color function obeying to its own transport equation to represent the phase quantity present in each element. One of the consequences of this method is the introduction of sharply varying coefficients in the equations solved, which could have a huge impact on the numerical error behaviour. It is indeed well-known that the convergence performances of standard multigrid schemes may be strongly impacted by the presence of discontinuities in the source terms and/or in the coefficients [41]. Another consequence is that the interface regularisation over a spatial span of size Δ itself implies a modelization error directly depending on Δ. Thus, an interesting perspective would be to analyze the impact of these additional sources of errors or uncertainties when performing mesh adaptation.

The compressible Navier-Stokes equations are solved using an All-Mach solver [111]. This solver is a projection-based numerical scheme, and thus it is mainly based on a projection step solving a Poisson-Helmholtz equation, as it is the case for the incompressible Navier-Stokes equations. It would be interesting to test whether the errors introduced in the projection step of the compressible solver can be also modeled as an extension of the error estimate proposed for incompressible Navier-Stokes equations in this thesis. Note that due to compressibility effects (shockwaves, ...), the performance of this estimate and the consequences in terms of error propagation of the solver must be clearly studied and identified.

Finally, the techniques developed should be applied to the investigation of real complex cases. One example is the problem of the collpase attached to a wall, which has been investigated within the framework of the Cachmap and Probalcav ANR project. The dynamic of this bubble depends on the contact angle α between the bubble and the wall: for $\alpha > \frac{\pi}{2}$, the standard cavitation jetting directed towards the wall is observed, whereas for $\alpha < \frac{\pi}{2}$ a jet opposed to the wall direction exists and has been observed experimentally. At first instants, this problem reduces to a Laplace equation for the pressure solved with mixed Dirichlet and Neumann boundary conditions, which, regarding the geometry, is known to present a regular solution with singular derivatives in the neighborhood of the corner formed by the bubble and the wall [145]. A similar type of solution is obtained for the well-known singular corner in potential flows [131]. Preliminary tests have shown that the total error can't be modelized through the local Hessian-based interpolation error, as numerical errors propagate from the singularity to the entire domain. As a consequence, the error model developed in this work is not sufficient for an accurate modelization of the numerical errors in problems where singularities are present and further investigation is required. Typically, a local error analysis should be defined in the vicinity of the singularity (similar to what is done for shock-dominated problems in compressible flows) and a more global one (for example using our proposed method) onto the rest of the domain.

Going back to the bubble collapse case, it may be shown that the Laplace problem for the pressure at initial time governs the oscillations of the bubbles. In particular, it may be demonstrated that the bubble resonance frequency depends on the gradient pressure integral over the bubble surface. Thus, a case study could have as objective to find this frequency and the use of a goal-oriented error estimate could be efficient to obtain a more accurate estimation of the output functional.

A | Error on a square element of reference

In this section, we derive the $\boldsymbol{L}^1$-norm interpolation error estimate for a square element K represented by its vertex $(\boldsymbol{v}_i)_{i\in[\![1,4]\!]}$. To do so, we show off an exact point-wise error estimate of the interpolation error within an element of reference K_{ref}. We apply a change of variables to obtain the error on K from the error on K_{ref}. Once it is obtained, the equivalent metric of the octree element K is deduced.

The reference element is a cubic element with unit edge lengths. Its vertices are noted $\hat{\boldsymbol{v}}_1 = (0,0)$, $\hat{\boldsymbol{v}}_2 = (1,0)$, $\hat{\boldsymbol{v}}_3 = (0,1)$ and $\hat{\boldsymbol{v}}_4 = (1,1)$.

The transformation between a coordinate $\boldsymbol{x} \in K$ and $\hat{\boldsymbol{x}} \in K_{ref}$ is given by

$$\boldsymbol{x} = \boldsymbol{v}_1 + h\hat{\boldsymbol{x}} \tag{A.1}$$

with h the edge length of K.

Let $u(\boldsymbol{x}) = \boldsymbol{x}^T H \boldsymbol{x}$ be a quadratic function represented by its associated Hessian $H = \begin{pmatrix} a & d \\ d & b \end{pmatrix}$ and exact-valued at the vertices of the cube. We pose Π_h the bilinear interpolate operator. In the framework of K_{ref}, u reads

$$u(\boldsymbol{x}(\hat{\boldsymbol{x}})) = \frac{1}{2}\boldsymbol{v}_1^T H \boldsymbol{v}_1 + \frac{1}{2}\boldsymbol{v}_1^T H h \hat{\boldsymbol{x}} + \frac{1}{2}\hat{\boldsymbol{x}}^T h H \boldsymbol{v}_1 + \frac{1}{2}h^2 \hat{\boldsymbol{x}}^T H \hat{\boldsymbol{x}} \ . \tag{A.2}$$

Linear and constant terms of $u(\boldsymbol{x}(\hat{bx}))$ are exactly interpolated, thus we consider, without loss in generality, only the quadratic term $\hat{u}(\boldsymbol{x}) = \frac{1}{2}h^2\hat{\boldsymbol{x}}^T H \hat{\boldsymbol{x}}$, as $(u - \Pi_h u)(\boldsymbol{x}) = (\hat{u} - \Pi_h \hat{u})(\boldsymbol{x})$. To improve readability, we write u instead of $\hat{u}$ in the following.

For all $\boldsymbol{x} = (x,y)^T \in K_{ref}$, the bilinear interpolate $\Pi_h u$ of the function u is

$$\Pi_h u(\boldsymbol{x}) = \alpha + \lambda x + \mu y + \beta xy \ . \tag{A.3}$$

As the solution is known at the vertices, we have

$$\begin{cases} \Pi_h u(\boldsymbol{v}_1) & = \alpha = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_1)) = 0 \\ \Pi_h u(\boldsymbol{v}_2) & = \lambda = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_2)) = \frac{1}{2}ah^2 \\ \Pi_h u(\boldsymbol{v}_3) & = \mu = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_3)) = \frac{1}{2}bh^2 \\ \Pi_h u(\boldsymbol{v}_4) & = \lambda + \mu + \beta = u(\boldsymbol{x}(\hat{\boldsymbol{v}}_4)) = \frac{1}{2}(a+b+2d)h^2 \end{cases}$$

From that, we readily obtain the point-wise interpolate

$$\Pi_h u(\boldsymbol{x}) = \frac{1}{2} a h^2 x + \frac{1}{2} b h^2 y + d h^2 x y \ . \tag{A.4}$$

and the point-wise interpolation error on the reference element K_{ref}

$$(u - \Pi_h u)(\boldsymbol{x}) = \frac{1}{2} h^2 \left[a(x^2 - x) + b(y^2 - y) \right] \ . \tag{A.5}$$

By direct integration, we obtain the interpolation error on K_{ref} in $\boldsymbol{L}^1$-norm

$$\begin{aligned} ||u - \Pi_h u||_{\boldsymbol{L}^1(K_{ref})} &= \int_0^1 \int_0^1 |u - \Pi_h u|(\boldsymbol{x}) dx\, dy & \text{(A.6)} \\ &= \frac{1}{12} \operatorname{tr}(H) h^2 & \text{(A.7)} \end{aligned}$$

We apply the change of variables between a coordinate $\boldsymbol{x} \in K$ and $\hat{\boldsymbol{x}} \in K_{ref}$. It writes

$$\int_K f(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} = \int_{K_{ref}} f(\hat{\boldsymbol{x}}) h^2 \mathrm{d}\hat{\boldsymbol{x}} = |K| \int_{K_{ref}} f(\hat{\boldsymbol{x}}) \mathrm{d}\hat{\boldsymbol{x}} \tag{A.8}$$

with $|K| = h^2$ the volume of the element K. Thus, the $\boldsymbol{L}^1$-norm interpolation error of the element K writes

$$||u - \Pi_h u||_{\boldsymbol{L}^1(K)} = \frac{1}{12} \operatorname{tr}(H) h^2 |K| \ . \tag{A.9}$$

B| Optimal quad/octree interpolation error proof

In this appendix, we detail the resolution of the quad/octree interpolation error minimization problem made in chapter 3 section 4.2.

The minimization problem writes :

$$\text{Find } \min_{\boldsymbol{M}} \left(\int_{\Omega} (|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}))^p \ \mathrm{d}\boldsymbol{x} \right) , \tag{B.1}$$

$$\text{under the constraint } \mathcal{C}(\boldsymbol{M}) = \int_{\Omega} (h^n(\boldsymbol{x}))^{-1} \ \mathrm{d}\boldsymbol{x} = N_*. \tag{B.2}$$

The resolution of the optimization problem is based on a change of variable. As the searched metric is isotropic, the anisotropic quotients r_i defined in the anisotropic case are no more necessary. Only the local density of the metric, define as $d = h^{-n}$, is used. After introducing this change of variable, we readily obtain the following minimization problem:

$$\text{Find } \min_{\boldsymbol{M}} \left(\int_{\Omega} \left(d^{-\frac{2}{n}}(\boldsymbol{x}) \sum_{i=1}^{n} \gamma_i(\boldsymbol{x}) \right)^p \ \mathrm{d}\boldsymbol{x} \right) , \tag{B.3}$$

$$\text{under the constraint } \mathcal{C}(\boldsymbol{M}) = \int_{\Omega} d(\boldsymbol{x}) \ \mathrm{d}\boldsymbol{x} = N_* \ . \tag{B.4}$$

Resolution

The problem is solved using Lagrange multipliers theory. This theory states a necessary condition verified by the solution of the optimal problem: the variation of the quantity to minimize $E_p = \int_{\Omega} \left(d^{-\frac{2}{n}}(\boldsymbol{x}) \sum_{i=1}^{n} \gamma_i(\boldsymbol{x}) \right)^p \ \mathrm{d}\boldsymbol{x}$ at point $\boldsymbol{M}$ in direction $\delta\boldsymbol{M}$ is proportional to the variation of the constraint at this point. In other words, it exists a unique real α such that

$$\delta E_p(\boldsymbol{M}, \delta\boldsymbol{M}) + \alpha \, \delta\mathcal{C}(\boldsymbol{M}, \delta\boldsymbol{M}) = 0 \ . \tag{B.5}$$

The variation of E_p is approximated by:

$$\delta E_p(\boldsymbol{M}, \delta\boldsymbol{M}) \approx \int_{\Omega} \frac{\partial}{\partial \mathcal{M}} \left(\left(d^{-\frac{2}{n}}(\boldsymbol{x}) \sum_{i=1}^{n} \gamma_i(\boldsymbol{x}) \right)^p \right) d\mathcal{M} \tag{B.6}$$

The constraint is constant, which implies that its variation is null:

$$\delta\mathcal{C}(\boldsymbol{M}, \delta\boldsymbol{M}) = 0 \tag{B.7}$$

The Lagrange multiplier condition reduces then to

$$\delta E_p(\boldsymbol{M}, \delta\boldsymbol{M}) = 0 \tag{B.8}$$

The variation of the metric is $\delta\boldsymbol{M} = (\delta d)$. It follows:

$$\frac{\partial}{\partial\mathcal{M}}\left(\left(d^{-\frac{2}{n}}(\boldsymbol{x})\sum_{i=1}^{n}\boldsymbol{\gamma}_i(\boldsymbol{x})\right)^p\right) = \frac{\partial}{\partial d}\left(\left(d^{-\frac{2}{n}}(\boldsymbol{x})\sum_{i=1}^{n}\boldsymbol{\gamma}_i(\boldsymbol{x})\right)^p\right) \tag{B.9}$$

$$= -\frac{2p}{n}\left(\sum_{i=1}^{n}\boldsymbol{\gamma}_i\right)^p d^{-\frac{2p+n}{n}} \ . \tag{B.10}$$

Equations B.8 and B.10 leads directly to

$$d = K\left(\sum_{i=1}^{n}\boldsymbol{\gamma}_i\right)^{\frac{np}{2p+n}} \ , \tag{B.11}$$

with K a constant.

Now the constraint $\mathcal{C}(\boldsymbol{M}) = N_*$ is applied to determine the constant K. From that, we readily obtain

$$d = N_*\left(\int_\Omega\left(\sum_{i=1}^{n}\boldsymbol{\gamma}_i\right)^{\frac{np}{2p+n}}\right)^{-1}\left(\sum_{i=1}^{n}\boldsymbol{\gamma}_i\right)^{\frac{np}{2p+n}} \tag{B.12}$$

Inverting the change of variable, we found the optimal size

$$h_{opt}(\boldsymbol{x}) = N_*^{-\frac{1}{n}}\left(\int_\Omega\left(\sum_{i=1}^{n}\boldsymbol{\gamma}_i(\boldsymbol{x})\right)^{\frac{np}{2p+n}}\mathrm{d}\boldsymbol{x}\right)^{\frac{1}{n}}\left(\sum_{i=1}^{n}\boldsymbol{\gamma}_i(\boldsymbol{x})\right)^{-\frac{p}{2p+n}} \ , \tag{B.13}$$

the point-wise local error

$$|u - \pi_{\mathcal{M}}u|(\boldsymbol{x}) = C_n N_*^{-\frac{2}{n}}\left(\int_\Omega\left(\sum_{i=1}^{n}\boldsymbol{\gamma}_i(\boldsymbol{x})\right)^{\frac{np}{2p+n}}\mathrm{d}\boldsymbol{x}\right)^{\frac{2}{n}}\left(\sum_{i=1}^{n}\boldsymbol{\gamma}_i(\boldsymbol{x})\right)^{\frac{n}{2p+n}} \ , \tag{B.14}$$

and the optimal global error

$$||u - \pi_{\mathcal{M}}u||_{\Omega,L^p} = C_n N_*^{-\frac{2}{n}}\left(\int_\Omega\left(\sum_{i=1}^{n}\boldsymbol{\gamma}_i(\boldsymbol{x})\right)^{\frac{np}{2p+n}}\mathrm{d}\boldsymbol{x}\right)^{\frac{2p+n}{np}} \ . \tag{B.15}$$

Uniqueness

In the previous paragraph, we found a solution of the optimization problem. Now, we prove that this solution is the unique solution which verifies $E_p(\boldsymbol{M}_{opt})^p \leq E_p(\boldsymbol{M})^p$ for all $\boldsymbol{M}$ having the same fixed $(\boldsymbol{\gamma}_i)_{i\in[\![1,n]\!]}$, and $\boldsymbol{M}_{opt}$ being the optimal metric previously found.

From equation (B.15), we have

$$E_p(\boldsymbol{M}_{opt})^p = C_n^p N_*^{-\frac{2p}{n}} \left(\int_\Omega \left(\sum_{i=1}^n \boldsymbol{\gamma}_i(\boldsymbol{x}) \right)^{\frac{np}{2p+n}} \, \mathrm{d}\boldsymbol{x} \right)^{\frac{2p+n}{n}} . \tag{B.16}$$

The density equation (B.12) is rewritten as $d = N_* (\int_\Omega f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x})^{-1} f$, where f is a strictly positive function. For any set of metric $\boldsymbol{M}$, the global error integrated from equation (3.75) writes

$$E_p(\boldsymbol{M})^p = C_n^p N_*^{-\frac{2p}{n}} \left(\int_\Omega f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \right)^{\frac{2p}{n}} \left(\int_\Omega f(\boldsymbol{x})^{-\frac{2p}{n}} \left(\sum_{i=1}^n \boldsymbol{\gamma}_i \right)^p \, \mathrm{d}\boldsymbol{x} \right) . \tag{B.17}$$

The inequality $(\sum_{i=1}^n \boldsymbol{\gamma}_i)^{\frac{np}{2p+n}} \leq (\sum_{i=1}^n \boldsymbol{\gamma}_i)^p$ is equivalent to the inequality $\frac{np}{2p+n} \leq p$ as $\frac{np}{2p+n} \geq 0$, $p \geq 0$ and $\boldsymbol{\gamma}_i \geq 0$, which is equivalent to $p^2 \geq 0$. Thus it is always verified. We introduce $g = (\sum_{i=1}^n \boldsymbol{\gamma}_i)^{\frac{np}{2p+n}}$. This leads to

$$E_p(\boldsymbol{M}_{opt})^{\frac{np}{2p+n}} = c_n^{\frac{np}{2p+n}} N_*^{-\frac{2p}{2p+n}} \int_\Omega g(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \tag{B.18}$$

and

$$E_p(\boldsymbol{M})^{\frac{np}{2p+n}} \geq C_n^{\frac{np}{2p+n}} N_*^{-\frac{2p}{2p+n}} \left(\int_\Omega f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \right)^{\frac{2p}{2p+n}} \left(\int_\Omega f(\boldsymbol{x})^{-\frac{2p}{n}} g(\boldsymbol{x})^{\frac{2p+n}{n}} \, \mathrm{d}\boldsymbol{x} \right)^{\frac{np}{2p+n}} . \tag{B.19}$$

Using Hölder inequality, it comes

$$\left(\int_\Omega f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \right)^{\frac{2p}{2p+n}} \left(\int_\Omega f(\boldsymbol{x})^{-\frac{2p}{n}} g(\boldsymbol{x})^{\frac{2p+n}{n}} \, \mathrm{d}\boldsymbol{x} \right)^{\frac{np}{2p+n}} = \left(\int_\Omega f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \right)^{\frac{2p}{2p+n}} \left(\int_\Omega \left(\frac{g(\boldsymbol{x})}{f(\boldsymbol{x})^{\frac{2p}{2p+n}}} \right)^{\frac{2p+n}{n}} \, \mathrm{d}\boldsymbol{x} \right)^{\frac{np}{2p+n}} \tag{B.20}$$

$$\geq \int_\Omega f(\boldsymbol{x})^{\frac{2p}{2p+n}} \left(\frac{g(\boldsymbol{x})}{f(\boldsymbol{x})^{\frac{2p}{2p+n}}} \right) \, \mathrm{d}\boldsymbol{x} \tag{B.21}$$

$$\geq \int_\Omega g(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \ , \tag{B.22}$$

as $\frac{2p+n}{2p} \geq 1$, $\frac{2p+n}{n} \geq 1$ and $\frac{2p}{2p+n} + \frac{n}{2p+n} = 1$. Equation (B.22) implies $E_p(\boldsymbol{M}_{opt})^p \leq E_p(\boldsymbol{M})^p$ for all $\boldsymbol{M}$ having the same fixed $(\boldsymbol{\gamma}_i)_{i \in [\![1,n]\!]}$. The initial optimization problem is strictly convex, so the optimal solution $\boldsymbol{M}_{opt}$ is unique.

Finally, it is easy to verify that the error is minimal when the eigen directions of the Hessian and the metric are aligned. From that, the optimal metric is deduced. In particular, the optimal size is given by

$$h_{opt}(\boldsymbol{x}) = N_*^{-\frac{1}{n}} \left(\int_\Omega (\, \mathrm{tr}(|H|(\boldsymbol{x})))^{\frac{np}{2p+n}} \, \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{n}} (\, \mathrm{tr}(|H|(\boldsymbol{x})))^{-\frac{p}{2p+n}} . \tag{B.23}$$

The point-wise local error is

$$|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) = C_n N_*^{-\frac{2}{n}} \left(\int_\Omega (\, \mathrm{tr}(|H|(\boldsymbol{x})))^{\frac{np}{2p+n}} \, \mathrm{d}\boldsymbol{x} \right)^{\frac{2}{n}} (\, \mathrm{tr}(|H|(\boldsymbol{x})))^{\frac{n}{2p+n}} \ , \tag{B.24}$$

with $C_2 = C_3 \frac{1}{12}$. The optimal global error is

$$||u - \pi_{\mathcal{M}} u||_{\Omega,L^p} = C_n N_*^{-\frac{2}{n}} \left(\int_\Omega \left(\operatorname{tr}(|H|(\boldsymbol{x})) \right)^{\frac{np}{2p+n}} \, \mathrm{d}\boldsymbol{x} \right)^{\frac{2p+n}{np}} , \tag{B.25}$$

where:

- N_* is the complexity ;
- n is the dimension of the problem ;
- p is the norm in which the error is minimized ;
- $|H|$ is the absolute Hessian of the sensor ;
- $\operatorname{tr}(A)$ denotes the trace of the matrix A.

From the definition of the complexity and introducing the relation (3.62) between the number of element N and the mean elements size h, we get

$$N_* = \int_\Omega (h(\boldsymbol{x})^{-n} \, \mathrm{d}\boldsymbol{x} = \frac{L_0^n}{h^n} = N \tag{B.26}$$

From that, the continuous error writes

$$|u - \pi_{\mathcal{M}} u|(\boldsymbol{x}) = C_n \, N^{-\frac{2}{n}} \left(\int_\Omega \left(\operatorname{tr}(|H|(\boldsymbol{x})) \right)^{\frac{np}{2p+n}} \, \mathrm{d}\boldsymbol{x} \right)^{\frac{2}{n}} \left(\operatorname{tr}(|H|(\boldsymbol{x})) \right)^{\frac{n}{2p+n}} , \tag{B.27}$$

with $C_2 = C_3 = \frac{1}{12}$, and

$$||u - \pi_{\mathcal{M}} u||_{\Omega,L^p} = C_n \, N^{-\frac{2}{n}} \left(\int_\Omega \left(\operatorname{tr}(|H|(\boldsymbol{x})) \right)^{\frac{np}{2p+n}} \, \mathrm{d}\boldsymbol{x} \right)^{\frac{2p+n}{np}} . \tag{B.28}$$

C| Problems of reference

In this chapter, we shortly introduce some reference solutions and problems regularly used during this thesis.

1 Multiscale sinusoidal problem

We name multiscale sinusoidal problem the problem solved by the solution $u = u_1$ with

$$u_1(x, y) = \sin(a\,(xy - b))\left(0.01 + e^{-(d\,(xy-c))^2}\right) \tag{C.1}$$

with $a = 5\pi$, $b = 1.5\pi/50$, $c = 3\pi/50$ and $d = 50\pi$. It is illustrated in figure C.1.

This choice is inspired from (but not identical to) [30, 46]. It is motivated by its multiscale aspect, with two scales in amplitude and two scales in wavelength. It combines an harmonic function with a decaying exponential function. This type of combination is typically encountered in the classical solutions of systems containing instabilities.

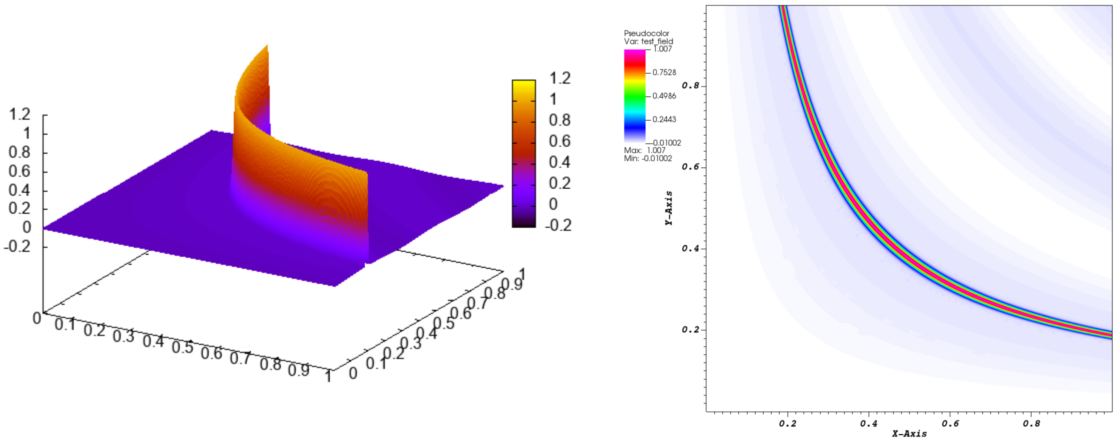


Figure C.1: Representation of the function $u_1(x, y)$.

When we solve the Poisson-Helmholtz equation

$$\nabla \cdot (D\nabla u) + \lambda u = s \tag{C.2}$$

with a user-prescribed diffusion coefficient D, reaction-coefficient λ and a source term s, s is computed such that the solution is $u = u_1$:

$$\begin{aligned}
s = &-D[\sin(a(xy-b))(ay)^2(0.01+\exp(-(d(xy-c))^2)) \\
&+ 4\cos(a(xy-b))a(yd)^2(xy-c)\exp(-(d(xy-c))^2) \\
&+ 2\sin(a(xy-b))(dy)^2\exp(-(d(xy-c))^2) \\
&- 4\sin(a(xy-b))d^4((xy-c)y)^2\exp(-(d(xy-c))^2) \\
&+ \sin(a(xy-b))(ax)^2(0.01+\exp(-(d(xy-c))^2)) \\
&+ 4\cos(a(xy-b))a(xd)^2(xy-c)\exp(-(d(xy-c))^2) \\
&+ 2\sin(a(xy-b))(dx)^2\exp(-(d(xy-c))^2) \\
&- 4\sin(a(xy-b))d^4((xy-c)x)^2\exp(-(d(xy-c))^2)] \\
&\lambda\sin(a(xy-b))(0.01+\exp(-(d(xy-c))^2))
\end{aligned}$$

2 Boundary layer problem

We name boundary layer problem the problem solved by the solution $u = u_2$ with

$$u_2(x,y) = xy^2 - y^2\exp(\frac{2(x-1)}{\kappa}) - x\exp(\frac{3(y-1)}{\kappa}) + \exp(\frac{2(x-1)+3(y-1)}{\kappa}) \ , \tag{C.3}$$

with $\kappa = 10^{-2}$. This solution represents a regular boundary layer and is presented by John in [126]. It is illustrated in figure C.2.

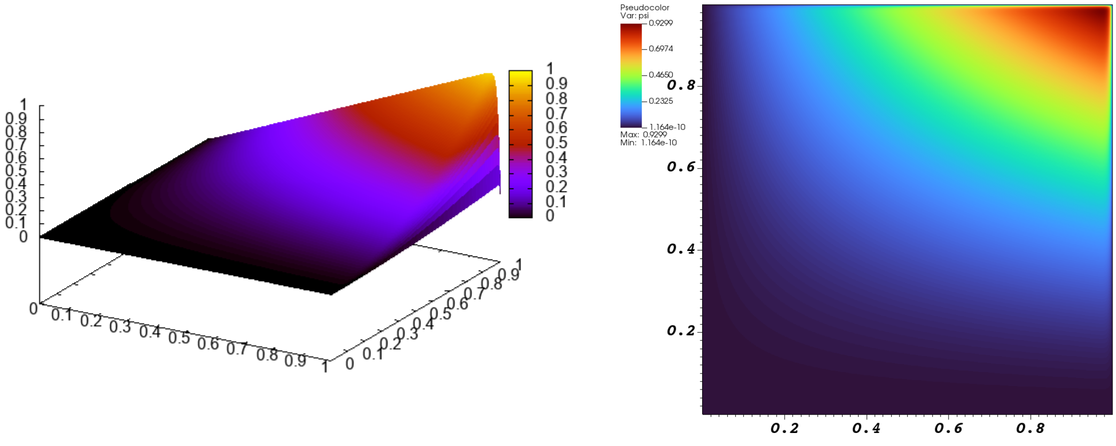


Figure C.2: Representation of the function $u_2(x, y)$.

When we solve the Poisson-Helmholtz equation

$$\nabla \cdot (D\nabla u) + \lambda u = s \tag{C.4}$$

with a user-prescribed diffusion coefficient D, reaction-coefficient λ and a source term s, s is computed such that the solution is $u = u_2$:

$$\begin{aligned}
s = {} & \lambda x y^2 \\
& + 2Dx \\
& + \exp((2(x-1))/\kappa)\left(-\lambda y^2 - \frac{4D}{\kappa^2} y^2 - 2D\right) \\
& + \exp((3(y-1))/\kappa)\left(-\lambda x - \frac{9D}{\kappa^2} x\right) \\
& + \exp((2(x-1)+3(y-1))/\kappa)\left(\lambda + \frac{13D}{\kappa^2}\right)
\end{aligned}$$

D | Poisson-Helmholtz solver – A refining study

In chapter 4, section 1, we perform an interpolation error based coarsening experiment which shows the importance of the mesh compression ratio to obtain an efficient AMR method. In particular, we observe that when coarsening a uniformly refined mesh where the interpolation error is minimal, two error regimes may be observed: in the first one, the interpolation and the total error are proportional, whereas in the second one they are both decorrelated. The limit between these two regimes are characterized by a problem dependant mesh compression ratio.

In this appendix, we show that this behaviour is similar on when performing a refining study, which is the opposite to the coarsening study: instead of coarsen a fine mesh, we refine an initially coarse cartesian grid where the interpolation error is the highest. This refining experiment is performed on the same numerical solution of the same Poisson-Helmholtz equation than the coarsening experiment (*cf.* chapter 4, section 1 and appendix C). The initial mesh contains element with an imposed minimum level $l_{min} = 5$, and the maximum level is set to different maximum level ($l_{max} \in [\![8, 9]\!]$).

The error convergence obtained is presented in figure D.1. We observe the presence of the two same error regimes and the error curves are very similar to the case of the coarsening study. This implies that the behaviour of the total error is similar in both case.

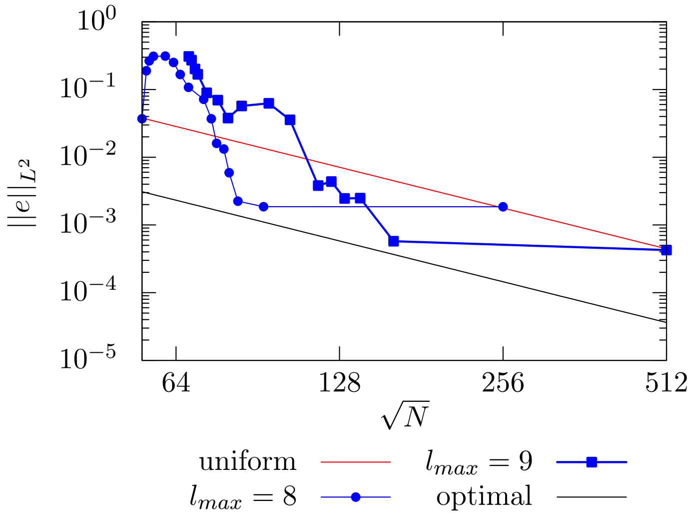
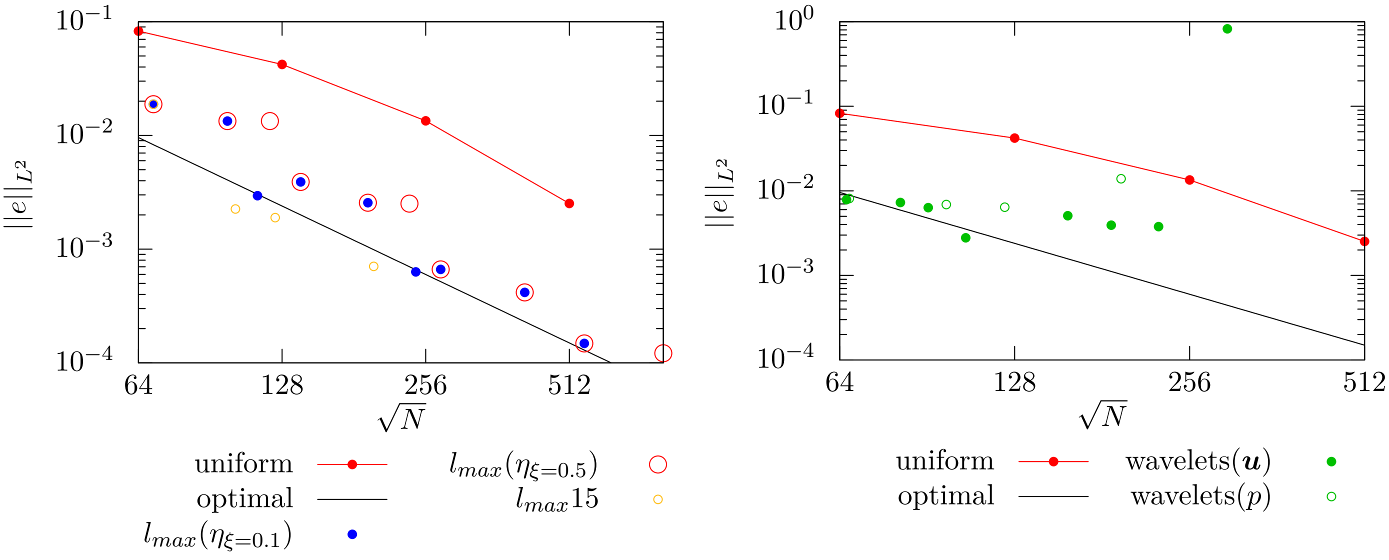


Figure D.1: Error convergence for the refining study: a coarse mesh (level $l = 5$) is refined where the interpolation error on the multiscale sinusoidal numerical solution is the highest.

E| Lamb-Oseen vortex additional results

In chapter 5, section 2.1, we describe the performance of our AMR method applied to a Lamb-Oseen vortex solution of the Euler equations for two values of the security coefficient ($\xi = 0.1$ and $\xi = 0.5$) and in comparison with the wavelet-based AMR method of Basilisk. In figure E.1, we show the results obtained in terms of error on the velocity field. The same conclusions than for the error on the pressure field arise: the wavelet-based AMR method does not reduce the error when the number of element increases, whereas the proposed method leads to a nearly optimal error convergence. The results obtained with a security coefficient $\xi = 0.1$ are closer to the optimal expectations than when a coefficient $\xi = 0.5$ is used.

Thus, the proposed method shows better performance than the wavelet-based method, and the it is controlled by the choice of the security coefficient.



(a) Comparison between the imposed minimal cell size constrain ($\xi = 0.1$, $\xi = 0.5$, no constrain).

(b) Comparison of performance using the wavelet-based method.

Figure E.1: error on $\boldsymbol{u}$ for the Lamb-Oseen vortex.

BIBLIOGRAPHY

[1] A. Berny, *Étude numérique de l'éclatement d'une bulle à la surface de différents liquides*. Theses, Sorbonne Université, Sept. 2020.

[2] H. Borouchaki, F. Hecht, and P. J. Frey, "Mesh gradation control," *International Journal for Numerical Methods in Engineering*, vol. 43, no. 6, pp. 1143–1165, 1998.

[3] A. M. Collier, K. G. A., M. D. Jones, K. Myers, and A. Jerusalem, "Numerical simulation of the onset of the second stage of labor," in *8th ECCOMAS Congress*, 2022.

[4] J. P. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. J. Mavriplis, "Cfd vision 2030 study: a path to revolutionary computational aerosciences," tech. rep., 2014.

[5] M. A. Park, A. Loseille, J. Krakos, T. R. Michal, and J. J. Alonso, "Unstructured grid adaptation: status, potential impacts, and recommended investments towards cfd 2030," in *46th AIAA fluid dynamics conference*, p. 3323, 2016.

[6] S. Popinet, "http://basilisk.fr/." Accessed: 2022-08-20.

[7] L. Bergamasco and D. Fuster, "Oscillation regimes of gas/vapor bubbles," *International Journal of Heat and Mass Transfer*, vol. 112, pp. 72–80, 2017.

[8] J. Bourguille, L. Bergamasco, G. Tahan, D. Fuster, and M. Arrigoni, "Shock propagation effects in multilayer assembly including a liquid phase," in *Structural and Mechanical Engineering for Security and Prevention*, vol. 755 of *Key Engineering Materials*, pp. 181–189, Trans Tech Publications Ltd, 10 2017.

[9] P. J. Roache, "Verification of codes and calculations," *AIAA Journal*, vol. 36, no. 5, pp. 696–702, 1998.

[10] W. L. Oberkampf and T. G. Trucano, "Verification and validation in computational fluid dynamics," *Progress in Aerospace Sciences*, vol. 38, no. 3, pp. 209–272, 2002.

[11] D. Pelletier, "Verification, validation, and uncertainty in computational fluids dynamicsthis article is one of a selection of papers published in this special issue on hydrotechnical engineering.," *Canadian Journal of Civil Engineering*, vol. 37, no. 7, pp. 1003–1013, 2010.

[12] F. Alauzet and L. Frazza, "Feature-based and goal-oriented anisotropic mesh adaptation for rans applications in aeronautics and aerospace," *Journal of Computational Physics*, vol. 439, p. 110340, 2021.

[13] N. Kowalski, F. Ledoux, and P. Frey, "Smoothness driven frame field generation for hexahedral meshing," *Computer-Aided Design*, vol. 72, pp. 65–77, 2016.

[14] N. Ray, D. Sokolov, M. Reberol, F. Ledoux, and B. Lévy, “Hex-dominant meshing: mind the gap!,” *Computer-Aided Design*, vol. 102, pp. 94–103, 2018.

[15] R. Lohner, “Advances in feflo,” in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 373, 2002.

[16] C. Dobrzynski and P. Frey, “Anisotropic delaunay mesh adaptation for unsteady simulations,” in *Proceedings of the 17th International Meshing Roundtable* (R. V. Garimella, ed.), (Berlin, Heidelberg), pp. 177–194, Springer Berlin Heidelberg, 2008.

[17] C. Geuzaine and J.-F. Remacle, “Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities,” *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.

[18] N. Pietroni, M. Campen, A. Sheffer, G. Cherchi, D. Bommes, X. Gao, R. Scateni, F. Ledoux, J.-F. Remacle, and M. Livesu, “Hex-mesh generation and processing: a survey,” *ACM Transactions on Graphics (TOG)*, 2022.

[19] R. Schneiders, “Octree-based hexahedral mesh generation,” *International Journal of Computational Geometry & Applications*, vol. 10, no. 04, pp. 383–398, 2000.

[20] A. Limare, *Raffinement adaptatif de maillages intersectants, en Volumes Finis d’ordre élevé, pour l’aéropropulsion.* PhD thesis, 2017. Thèse de doctorat dirigée par Borouchaki, Houman Matériaux Mécanique Optique et Nanothechnologie Troyes 2017.

[21] K. Schmidmayer, F. Petitpas, and E. Daniel, “Adaptive mesh refinement algorithm based on dual trees for cells and faces for multiphase compressible flows,” *Journal of Computational Physics*, vol. 388, pp. 252–278, 2019.

[22] H.-S. Oh and I. Babuška, “The p-version of the finite element method for the elliptic boundary value problems with interfaces,” *Computer Methods in Applied Mechanics and Engineering*, vol. 97, no. 2, pp. 211–231, 1992.

[23] S. Ghosh and S. K. Manna, “R-adapted arbitrary lagrangian-eulerian finite-element method in metal-forming simulation,” *Journal of Materials Engineering and Performance*, vol. 2, pp. 271–282, Apr 1993.

[24] A. Rajagopal and S. M. Sivakumar, “A combined r-h adaptive strategy based on material forces and error assessment for plane problems and bimaterial interfaces,” *Computational Mechanics*, vol. 41, pp. 49–72, Dec 2007.

[25] S. Popinet, “Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries,” *Journal of Computational Physics*, vol. 190, no. 2, pp. 572–600, 2003.

[26] J. A. Van Hooft, S. Popinet, C. C. Van Heerwaarden, S. J. A. Van Der Linden, S. R. De Roode, and B. J. H. Van de Wiel, “Towards Adaptive Grids for Atmospheric Boundary-Layer Simulations,” *Boundary-Layer Meteorology*, pp. 1–23, Feb. 2018.

[27] M. J. Berger and J. Oliger, “Adaptive mesh refinement for hyperbolic partial differential equations,” *Journal of Computational Physics*, vol. 53, no. 3, pp. 484–512, 1984.

[28] M. Berger and P. Colella, “Local adaptive mesh refinement for shock hydrodynamics,” *Journal of Computational Physics*, vol. 82, no. 1, pp. 64–84, 1989.

[29] X. Gao and C. P. Groth, “A parallel solution – adaptive method for three-dimensional turbulent non-premixed combusting flows,” *Journal of Computational Physics*, vol. 229, no. 9, pp. 3250–3275, 2010.

[30] F. Alauzet and A. Loseille, “A decade of progress on anisotropic mesh adaptation for computational fluid dynamics,” *Computer-Aided Design*, vol. 72, pp. 13–39, 2016. 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[31] X. Li, J.-F. Remacle, N. Chevaugeon, and M. S. Shephard, “Anisotropic mesh gradation control.,” in *IMR*, pp. 401–412, 2004.

[32] A. Loseille, A. Dervieux, and F. Alauzet, “Fully anisotropic goal-oriented mesh adaptation for 3d steady euler equations,” *Journal of Computational Physics*, vol. 229, no. 8, pp. 2866–2897, 2010.

[33] A. Belme, F. Alauzet, and A. Dervieux, “An a priori anisotropic goal-oriented error estimate for viscous compressible flow and application to mesh adaptation,” *Journal of Computational Physics*, vol. 376, pp. 1051–1088, 2019.

[34] J. Wackers, G. Deng, C. Raymond, E. Guilmineau, A. Leroyer, P. Queutey, and M. Visonneau, “Adaptive grid refinement for ship resistance computations,” *Ocean Engineering*, vol. 250, p. 110969, 2022.

[35] A. Belme, A. Dervieux, and F. Alauzet, “Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows,” *Journal of Computational Physics*, vol. 231, no. 19, pp. 6323–6348, 2012.

[36] J. Wackers, G. Deng, E. Guilmineau, A. Leroyer, P. Queutey, M. Visonneau, A. Palmieri, and A. Liverani, “Can adaptive grid refinement produce grid-independent solutions for incompressible flows?,” *Journal of Computational Physics*, vol. 344, pp. 364–380, 2017.

[37] F. B. Barros, S. P. B. Proença, and C. S. d. Barcellos, “Generalized finite element method in structural nonlinear analysis – a p-adaptive strategy,” *Computational Mechanics*, vol. 33, pp. 95–107, Jan 2004.

[38] J. Dannenhoffer and J. Baron, *Adaptive procedure for steady state solution of hyperbolic equations.*

[39] M. J. Aftosmis, “Upwind method for simulation of viscous flow on adaptively refined meshes,” *AIAA Journal*, vol. 32, no. 2, pp. 268–277, 1994.

[40] S. M. Mitran, “A comparison of adaptive mesh refinement approaches for large eddy simulation,” tech. rep., Washington univ. Seattle dept. of applied mathematics, 2001.

[41] S. Popinet, “An accurate adaptive solver for surface-tension-driven interfacial flows,” *Journal of Computational Physics*, vol. 228, pp. 5838 – 5866, 2009.

[42] G. Agbaglah, S. Delaux, D. Fuster, J. Hoepffner, C. Josserand, S. Popinet, P. Ray, R. Scardovelli, and S. Zaleski, “Parallel simulation of multiphase flows using octree adaptivity and the volume-of-fluid method,” *Comptes Rendus Mécanique*, vol. 339, no. 2, pp. 194–207, 2011. High Performance Computing.

[43] J. Wackers, G. Deng, E. Guilmineau, A. Leroyer, P. Queutey, and M. Visonneau, "Combined refinement criteria for anisotropic grid refinement in free-surface flow simulation," *Computers and Fluids*, vol. 92, pp. 209–222, 2014.

[44] J. Peraire, M. Vahdati, K. Morgan, and O. Zienkiewicz, "Adaptive remeshing for compressible flow computations," *Journal of Computational Physics*, vol. 72, no. 2, pp. 449–466, 1987.

[45] F. Hecht, B. Mohammadi, F. Hecht, and B. Mohammadi, *Mesh adaption by metric control for multi-scale phenomena and turbulence.*

[46] A. Loseille, *Adaptation de maillage anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides : Application à la prédiction haute-fidélité du bang sonique.* PhD thesis, 2008. Thèse de doctorat dirigée par Alauzet, Frédéric et Dervieux, Alain Mathématiques appliquées Paris 6 2008.

[47] L. F. Richardson and R. T. Glazebrook, "Ix. the approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 210, no. 459-470, pp. 307–357, 1911.

[48] L. Richardson and A. Gaunt, "The deferred approach to the limit. part i. single lattic. part ll. interpenetrationg lattices," *The Philosophical Transcations of the Royal Society of London. Series A, Containing papers of a Mathematical or Physical Character*, vol. 116, no. 3, pp. 405–41, 1927.

[49] P. J. Roache and P. M. Knupp, "Completed richardson extrapolation," *Communications in Numerical Methods in Engineering*, vol. 9, no. 5, pp. 365–374, 1993.

[50] E. Fehlberg, *Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems*, vol. 315. National aeronautics and space administration, 1969.

[51] O. C. Zienkiewicz and J. Z. Zhu, "A simple error estimator and adaptive procedure for practical engineerng analysis," *International Journal for Numerical Methods in Engineering*, vol. 24, no. 2, pp. 337–357, 1987.

[52] V. Dolejší and P. Solin, "hp-discontinuous galerkin method based on local higher order reconstruction," *Applied Mathematics and Computation*, vol. 279, pp. 219–235, 2016.

[53] J. Wackers, G. Deng, A. Leroyer, P. Queutey, and M. Visonneau, "Adaptive grid refinement for hydrodynamic flows," *Computers and Fluids*, vol. 55, pp. 85–100, 2012.

[54] K. MacLean and S. Nadarajah, "Anisotropic mesh generation and adaptation for quads using the lp-cvt method," *Journal of Computational Physics*, p. 111578, 2022.

[55] T. C. Baudouin, J.-F. Remacle, E. Marchandise, J. Lambrechts, and F. Henrotte, "Lloyd's energy minimization in the l p norm for quadrilateral surface mesh generation," *Engineering with Computers*, vol. 30, no. 1, pp. 97–110, 2014.

[56] B. Lévy and Y. Liu, "L p centroidal voronoi tessellation and its applications," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–11, 2010.

[57] C. Roy, *Review of Discretization Error Estimators in Scientific Computing.*

[58] I. Babuška and W. C. Rheinboldt, “A-posteriori error estimates for the finite element method,” *International Journal for Numerical Methods in Engineering*, vol. 12, no. 10, pp. 1597–1615, 1978.

[59] M. Ainsworth and J. Oden, “A posteriori error estimation in finite element analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 142, no. 1, pp. 1–88, 1997.

[60] R. Verfürth, “A posteriori error estimators for convection-diffusion equations,” *Numerische Mathematik*, vol. 80, pp. 641–663, Oct 1998.

[61] Vohralík, Martin, “Residual flux-based a posteriori error estimates for finite volume and related locally conservative methods,” *Numerische Mathematik*, vol. 111, pp. 121–158, 2008.

[62] Di Pietro, Daniele A. and Vohralík, Martin, “A review of recent advances in discretization methods, a posteriori error analysis, and adaptive algorithms for numerical modeling in geosciences,” *Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles*, vol. 69, no. 4, pp. 701–729, 2014.

[63] J. Banks, J. Hittinger, J. Connors, and C. Woodward, “Numerical error estimation for nonlinear hyperbolic pdes via nonlinear error transport,” *Computer Methods in Applied Mechanics and Engineering*, vol. 213-216, pp. 1–15, 2012.

[64] Y. Qin, T. Shih, P. Keller, R. Sun, E. Hernandez, C. Perng, N. Trigui, Z. Han, F. Shen, and T. Shieh, *Estimating Grid-Induced Errors in CFD by Discrete-Error-Transport Equations.*

[65] I. Celik and G. Hu, “Single Grid Error Estimation Using Error Transport Equation ,” *Journal of Fluids Engineering*, vol. 126, pp. 778–790, 12 2004.

[66] A. Hay and M. Visonneau, “Error estimation using the error transport equation for finite-volume methods and arbitrary meshes,” *International Journal of Computational Fluid Dynamics*, vol. 20, no. 7, pp. 463–479, 2006.

[67] J. M. Derlaga and M. A. Park, *Application of Exact Error Transport Equations and Adjoint Error Estimation to AIAA Workshops.*

[68] I. Babuška, T. Strouboulis, S. Gangaraj, and C. Upadhyay, “Pollution error in the h-version of the finite element method and the local quality of the recovered derivatives,” *Computer Methods in Applied Mechanics and Engineering*, vol. 140, no. 1, pp. 1–37, 1997.

[69] O. C. Zienkiewicz, “The background of error estimation and adaptivity in finite element computations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 4, pp. 207–213, 2006. Adaptive Modeling and Simulation.

[70] R. Becker and R. Rannacher, *Weighted A Posteriori Error Control in FE Methods.* IWR, 1996.

[71] R. Balasubramanian and J. C. Newman III, “Comparison of adjoint-based and feature-based grid adaptation for functional outputs,” *International Journal for Numerical Methods in Fluids*, vol. 53, no. 10, pp. 1541–1569, 2007.

[72] K. J. Fidkowski and D. L. Darmofal, “Review of output-based error estimation and mesh adaptation in computational fluid dynamics,” *AIAA Journal*, vol. 49, no. 4, pp. 673–694, 2011.

[73] N. A. Pierce and M. B. Giles, “Adjoint and defect error bounding and correction for functional estimates,” *Journal of Computational Physics*, vol. 200, no. 2, pp. 769–794, 2004.

[74] D. A. Venditti and D. L. Darmofal, “Grid adaptation for functional outputs: Application to two-dimensional inviscid flows,” *Journal of Computational Physics*, vol. 176, no. 1, pp. 40–69, 2002.

[75] R. Hartmann and P. Houston, “Adaptive discontinuous galerkin finite element methods for the compressible euler equations,” *Journal of Computational Physics*, vol. 183, no. 2, pp. 508–532, 2002.

[76] R. Hartmann, “Multitarget error estimation and adaptivity in aerodynamic flow simulations,” *SIAM Journal on Scientific Computing*, vol. 31, no. 1, pp. 708–731, 2008.

[77] G. Brèthes and A. Dervieux, “Anisotropic norm-oriented mesh adaptation for a poisson problem,” *Journal of Computational Physics*, vol. 322, pp. 804–826, 2016.

[78] A. Loseille, A. Dervieux, and F. Alauzet, *Anisotropic Norm-Oriented Mesh Adaptation for Compressible Flows.*

[79] L. Frazza, *3D anisotropic mesh adaptation for Reynolds Averaged Navier-Stokes simulations.* PhD thesis, 2018. Thèse de doctorat dirigée par Alauzet, Frédéric et Loseille, Adrien Mathématiques appliquées Sorbonne université 2018.

[80] L.-M. Tenkes and F. Alauzet, “Size gradation control for anisotropic hybrid meshes,” in *Numerical Geometry, Grid Generation and Scientific Computing* (V. A. Garanzha, L. Kamenski, and H. Si, eds.), (Cham), pp. 127–139, Springer International Publishing, 2021.

[81] R. Löhner, “Adaptive remeshing for transient problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 75, no. 1, pp. 195–214, 1989.

[82] F. Alauzet, “Size gradation control of anisotropic meshes,” *Finite Elements in Analysis and Design*, vol. 46, no. 1, pp. 181–202, 2010. Mesh Generation - Applications and Adaptation.

[83] Z. Xiao and C. F. O. Gooch, *Smooth Gradation of Anisotropic Mesh Based on Log-Euclidean Metrics.*

[84] K. Schneider and O. V. Vasilyev, “Wavelet methods in computational fluid dynamics,” *Annual Review of Fluid Mechanics*, vol. 42, no. 1, pp. 473–503, 2010.

[85] S. Popinet, “A quadtree-adaptive multigrid solver for the serre–green–naghdi equations,” *Journal of Computational Physics*, vol. 302, pp. 336–358, 2015.

[86] D. Fuster and S. Popinet, “An all-mach method for the simulation of bubble dynamics problems in the presence of surface tension,” *Journal of Computational Physics*, vol. 374, pp. 752–768, 2018.

[87] S. Popinet, "Quadtree-adaptive tsunami modelling," *Ocean Dynamics*, vol. 61, pp. 1261–1285, Sep 2011.

[88] S. Popinet, "http://basilisk.fr/src/readme." Accessed: 2022-08-20.

[89] F. Hecht, "Bamg: bidimensional anisotropic mesh generator," *User Guide. INRIA, Rocquencourt*, vol. 17, 1998.

[90] P. Frey, "YAMS A fully Automatic Adaptive Isotropic Surface Remeshing Procedure," Tech. Rep. RT-0252, INRIA, Nov. 2001.

[91] *STAR CCM+ Users Manual.*

[92] D. Boffi, M. Costabel, M. Dauge, and L. Demkowicz, "Discrete compactness for the hp version of rectangular edge finite elements," *SIAM Journal on Numerical Analysis*, vol. 44, p. 979–1004, 01 2006.

[93] G. Bessette, E. Becker, L. Taylor, and D. Littlefield, "Modeling of impact problems using an h-adaptive, explicit lagrangian finite element method in three dimensions," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 13, pp. 1649–1679, 2003.

[94] W. Hackbusch, *Multi-Grid Methods and Applications.* Springer, Berlin, Heidelberg, 1985.

[95] D. Martin and P. Colella, "A cell-centered adaptive projection method for the incompressible euler equations," *Journal of Computational Physics*, vol. 163, pp. 271–312, 06 2000.

[96] A. S. Almgren, J. B. Bell, P. Colella, and T. Marthaler, "A cartesian grid projection method for the incompressible euler equations in complex geometries," *SIAM Journal on Scientific Computing*, vol. 18, no. 5, pp. 1289–1309, 1997.

[97] A. J. Chorin, "Numerical solution of the Navier-Stokes equations," *Math. Comp.*, vol. 22, pp. 745–762, 1968.

[98] A. J. Chorin, "On the convergence of discrete approximations to the Navier-Stokes equations," *Math. Comp.*, vol. 23, pp. 341–353, 1969.

[99] T. D. T. a. Roger Peyret, *Computational Methods for Fluid Flow.* Scientific Computation, Springer Berlin Heidelberg, 1983.

[100] D. L. Brown, R. Cortez, and M. L. Minion, "Accurate projection methods for the incompressible navier–stokes equations," *Journal of Computational Physics*, vol. 168, no. 2, pp. 464–499, 2001.

[101] P.-Y. Lagrée, L. Staron, and S. Popinet, "The granular column collapse as a continuum: validity of a two-dimensional navier–stokes model with a mu(i)-rheology," *Journal of Fluid Mechanics*, vol. 686, p. 378–408, 2011.

[102] S. Popinet, "http://basilisk.fr/src/navier-stokes/centered.h." Accessed: 2022-08-20.

[103] J. B. Bell, P. Colella, and H. M. Glaz, "A second-order projection method for the incompressible navier-stokes equations," *Journal of Computational Physics*, vol. 85, no. 2, pp. 257–283, 1989.

[104] B. L. Bihari and A. Harten, “Multiresolution schemes for the numerical solution of 2-d conservation laws i,” *SIAM J. Sci. Comput.*, vol. 18(2), p. 315–354, 1997.

[105] B. L. Bihari, “Towards fast unstructured mesh simulations: Multiresolution methods and transactional memory.” Lecture notes for the12 th School on Computational Fluid Dynamics Roscoff, France, June 5-11, 2011, 2011. Lecture notes for the12 th School on Computational Fluid Dynamics Roscoff, France.

[106] B. Jawerth and W. Sweldens, “An overview of wavelet based multiresolution analyses,” *SIAM Rev.*, vol. 36(3), p. 377–412, 1994.

[107] A. Harten, “Adaptive multiresolution schemes for shock computations,” *Journal of Computational Physics*, vol. 115, no. 2, pp. 319–338, 1994.

[108] A. Harten, “Multiresolution algorithms for the numerical solution of hyperbolic conservation laws,” *Communications on Pure and Applied Mathematics*, vol. 48, no. 12, pp. 1305–1342, 1995.

[109] A. Cohen, S. Kaber, S. Müller, and M. Postel, “Fully adaptive multiresolution finite volume schemes for conservation laws,” *Mathematics of Computation*, vol. 72, no. 241, pp. 183–225, 2003.

[110] C. Ozhan, D. Fuster, and P. Da Costa, “Multi-scale flow simulation of automotive catalytic converters,” *Chemical Engineering Science*, vol. 116, pp. 161–171, 2014.

[111] D. Fuster, A. Bagué, T. Boeck, L. Le Moyne, A. Leboissetier, S. Popinet, P. Ray, R. Scardovelli, and S. Zaleski, “Simulation of primary atomization with an octree adaptive mesh refinement and vof method,” *International Journal of Multiphase Flow*, vol. 35, no. 6, pp. 550–565, 2009.

[112] A. Bagué, D. Fuster, S. Popinet, R. Scardovelli, and S. Zaleski, “Instability growth rate of two-phase mixing layers from a linear eigenvalue problem and an initial-value problem,” *Physics of Fluids*, vol. 22, no. 9, p. 092104, 2010.

[113] A. Loseille, A. Dervieux, P. Frey, and F. Alauzet, *Achievement of Global Second Order Mesh Convergence for Discontinuous Flows with Adapted Unstructured Meshes.*

[114] M. A. Park, A. Balan, W. K. Anderson, M. C. Galbraith, P. Caplan, H. A. Carson, T. R. Michal, J. A. Krakos, D. S. Kamenetskiy, A. Loseille, F. Alauzet, L. Frazza, and N. Barral, *Verification of Unstructured Grid Adaptation Components.*

[115] P. George, F. Hecht, and M. Vallet, “Creation of internal points in voronoi’s type method. control adaptation,” *Advances in Engineering Software and Workstations*, vol. 13, no. 5, pp. 303–312, 1991.

[116] M.-G. Vallet, *Génération de maillages éléments finis anisotropes et adaptatifs.* PhD thesis, 1992. Thèse de doctorat dirigée par Pironneau, Olivier Mathématiques Paris 6 1992.

[117] F. Alauzet and A. Loseille, “High-order sonic boom modeling based on adaptive methods,” *Journal of Computational Physics*, vol. 229, no. 3, pp. 561–593, 2010.

[118] P. L. George and *et al.*, *Meshing, Geometric Modeling and Numerical Simulation 2.* ISTE Ltd, 2019.

[119] B. M., *Geometry 1 and 2.* Universitext, Springer Verlag, 1987.

[120] B. M., *A Panoramic View of Riemannian Geometry.* Springer-Verlag Berlin Heidelberg, 2003.

[121] M. Do Carmo, *Differential Geometry of Curves and Surfaces.* Prentice-Hall, 1976.

[122] A. Loseille and F. Alauzet, "Continuous mesh framework part i: Well-posed continuous interpolation error," *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, pp. 38–60, 2011.

[123] A. Loseille and F. Alauzet, "Continuous mesh framework part ii: Validations and applications," *SIAM Journal on Numerical Analysis*, vol. 49, no. 1, pp. 61–86, 2011.

[124] D. Pagnutti and C. Ollivier-Gooch, "A generalized framework for high order anisotropic mesh adaptation," *Computers and Structures*, vol. 87, no. 11, pp. 670–679, 2009. Fifth MIT Conference on Computational Fluid and Solid Mechanics.

[125] M. D. Piggott, P. E. Farrell, C. R. Wilson, G. J. Gorman, and C. C. Pain, "Anisotropic mesh adaptivity for multi-scale ocean modelling," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1907, pp. 4591–4611, 2009.

[126] V. John, "A numerical study of a posteriori error estimators for convection–diffusion equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 5, pp. 757–781, 2000.

[127] G. Hauke, D. Fuster, and M. H. Doweidar, "Variational multiscale a-posteriori error estimation for multi-dimensional transport problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 33, pp. 2701–2718, 2008.

[128] P. G. Ciarlet, "Basic error estimates for elliptic problems," 1991.

[129] S. C. Brenner, L. R. Scott, and L. R. Scott, *The mathematical theory of finite element methods*, vol. 3. Springer, 2008.

[130] P.-J. Frey and F. Alauzet, "Anisotropic mesh adaptation for cfd computations," *Computer methods in applied mechanics and engineering*, vol. 194, no. 48-49, pp. 5068–5082, 2005.

[131] G. K. Batchelor, *An Introduction to Fluid Dynamics.* Cambridge Mathematical Library, Cambridge University Press, 2000.

[132] P. Kundu and I. Cohen, *Fluid Mechanics 2nd Edition.* Elsevier Exclusive, 2003.

[133] P. Kundu and I. Cohen, *Fluid Mechanics 2nd Edition*, ch. 5, p. 135. Elsevier Exclusive, 2003.

[134] K. Raghavan and M. Bernitsas, "Experimental investigation of reynolds number effect on vortex induced vibration of rigid circular cylinder on elastic supports," *Ocean Engineering*, vol. 38, no. 5, pp. 719–731, 2011.

[135] R. Stringer, J. Zang, and A. Hillis, "Unsteady rans computations of flow around a circular cylinder for a wide range of reynolds numbers," *Ocean Engineering*, vol. 87, pp. 1–9, 2014.

[136] E. Achenbach, “Distribution of local pressure and skin friction around a circular cylinder in cross-flow up to re= 5× 106,” *Journal of Fluid Mechanics*, vol. 34, no. 4, pp. 625–639, 1968.

[137] E. Achenbach and E. Heinecke, “On vortex shedding from smooth and rough cylinders in the range of reynolds numbers 6× 103 to 5× 106,” *Journal of fluid mechanics*, vol. 109, pp. 239–251, 1981.

[138] M. Zdravkovich, “Conceptual overview of laminar and turbulent flows past smooth and rough circular cylinders,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 33, no. 1, pp. 53–62, 1990.

[139] R. Gautier, D. Biau, and E. Lamballais, “A reference solution of the flow over a circular cylinder at re=40,” *Computers and Fluids*, vol. 75, pp. 103–111, 2013.

[140] U. Fey, M. König, and H. Eckelmann, “A new strouhal–reynolds-number relationship for the circular cylinder in the range 47< re< 2× 10 5,” *Physics of Fluids*, vol. 10, no. 7, pp. 1547–1549, 1998.

[141] U. Ghia, K. N. Ghia, and C. Shin, “High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method,” *Journal of computational physics*, vol. 48, no. 3, pp. 387–411, 1982.

[142] K. Ghia, J. W. Hankey, and J. Hodge, *Study of incompressible Navier-Stokes equations in primitive variables using implicit numerical technique.*

[143] P. Shankar and M. Deshpande, “Fluid mechanics in the driven cavity,” *Annual review of fluid mechanics*, vol. 32, no. 1, pp. 93–136, 2000.

[144] H. C. Kuhlmann and F. Romanò, “The lid-driven cavity,” *Computational Modelling of Bifurcations and Instabilities in Fluid Dynamics*, pp. 233–309, 2019.

[145] M. Costabel and M. Dauge, “Singularities of electromagnetic fields in polyhedral domains,” *Archive for Rational Mechanics and Analysis*, vol. 151, no. 3, pp. 221–276, 2000.